POPULAR
Computing
WEEKLY
28 October 1982 Vol 1 No 28
35p
FREE COMPUTER BUY/SELL SERVICE
COMPUTER SWAP
SEE PAGE 24
Timex-Sinclair 1000: The new micro that is sweeping the US
BBC Beebstick
Spectrum plot and draw
ZX81 memory miser
Cover Story: Halloween on BBC by Jeremy Ruston
Win a Jupiter Ace in our Better than Basic competition

VIDEO SOFTWARE LTD
NEW VIDEO-INDEX NEW
Q How do you store 57 000 characters in a 16K RAM?
A Use VIDEO-INDEX
Catalogue your cassettes, your record collection, references to magazine articles. Amazingly versatile. Capacity for 1 000 individual references each of which can generate 57 characters of text. Powerful search facilities once the index has been created.
How is it done?
The system cannot really fit 57 000 characters in a 16K RAM. It gives the illusion of doing so by eliminating duplicates. For instance in an index of articles in computer magazines about the ZX81 certain words occur time and again. VIDEO-INDEX detects these duplications and thereby conserves space.
What do you get?
1 A machine code master program which sits at the top of RAM. This is initially loaded like a BASIC program.
2 A demonstration index containing 1 000 references to articles about the ZX81 in the popular magazines.
3 A detailed instruction manual.
You may then proceed to create your own catalogues and indexes and save them on cassette. This program is fast, efficient and ingenious and is by far the best product we have introduced for the ZX81.
N.B. There isn't room for a commentary on the cassette so you have to read the manual. ZX81 or ZX80+8K ROM. ZX printer is useful but not essential. Price £9.95
STONE LANE
KINVER
STOURBRIDGE
WEST MIDLANDS
DY7 6EQ
ENGLAND
VIDEO
SOFTWARE LIMITED
PIMANIA
THE ADVENTURE GAME THAT'S FOR REAL !!!!
48K ZX SPECTRUM
MICHAEL ORWIN'S ZX81 CASSETTES
The best software (by various authors) at low prices
QUOTES
"Michael Orwin's £5 Cassette Two is very good value. It contains 10 solid well designed games which work, offer plenty of variety and choice, and are fun."
from the ZX Software review in Your Computer, May '82 issue
"I had your Invaders/React cassette ... I was delighted with this first cassette."
P Rubython, London NW10
"I have been intending to write to you for some days to say how much I enjoy the games on Cassette One, which you supplied me with earlier this month."
E H London SW4
"... I previously bought your Cassette One and consider it to be good value for money."
Richard Ross Langley
Managing Director
Mine of Information Ltd
CASSETTE 1
(eleven 1k programs)
machine code
React, Invaders, Phantom aliens, Maze of death, Planet lander, Bouncing letters, Bug splat
Basic
IChing, Mastermind, Robots, Basic Hangman
PLUS Large screen versions of Invaders and Maze of Death, ready for when you get 16k
Cassette 1 costs £3.80
CASSETTE 2
Ten games in Basic for 16k ZX81
Cassette Two contains Reversi, Awari, Laser Bases, Word Mastermind, Rectangles, Crash, Roulette, Pontoon, Penny Shoot and Gun Command.
Cassette 2 costs £5
CASSETTE 3
8 programs for 16k ZX81
STARSHIP TROJAN
Repair your Starship before disaster strikes. Hazards include asphyxiation, radiation, escaped biological specimens and plunging into a Supernova.
STARTREK This version of the well known space adventure game features variable Klingon mobility, and graphic photo torpedo tracking.
PRINCESS OF KRAAL
An adventure game
BATTLE Strategy game for 1 to 4 players
KALABRIASZ World's silliest card game, full of pointless complicated rules
CUBE Rubik Cube simulator, with lots of functions including 'Backstep'
SECRET MESSAGES This message coding program is very ...
MARTIAN CRICKET A simple but addictive game (totally unlike Earth cricket) in machine code. The speed is variable, and its top speed is very fast.
Cassette 3 costs £5
CASSETTE 4
8 games for 16k
ZX SCRAMBLE (machine code)
Bomb and shoot your way through the fortified caves.
GUNFIGHT
(machine code)
INVADERS
(machine code)
FUNGALOIDS (machine code)
GALAXY INVADERS (machine code)
Fleets of swooping and diving alien craft.
SNAKEBITE (machine code)
Eat the snake before it eats you. Variable speed (very fast at top speed)
LIFE (machine code)
A ZX81 version of the well known game
3D TIC TAC TOE (Basic)
Played on a 4×4×4 board, this is a game for the brain. It is very hard to beat the computer at it.
7 of the 8 games are in machine code, because this is much faster than Basic. (Some of these games were previously available from J. Steadman)
Cassette 4 costs £5
Recorded on quality cassettes, sent by first class post, from:
Michael Orwin, Dept P.O.C., 26 Brownlow Rd, Willesden, London NW10 9QL (mail order only please)
SPECTRUM SOFTWARE WANTED

POPULAR Computing WEEKLY

The Team

Editor
Brandon Gore

Reporter
David Kelly [01-930 3271]

Sub-editor
Ninette Sharp

Editorial Secretary
Theresa Lacy

Advertisement Manager
David Lake [01-839 2846]

Advertisement Executive
Alastair Macintosh [01-930 3840]

Managing Editor
Duncan Scot

Publishing Director
Jenny Ireland

Popular Computing Weekly,
Hobhouse Court, 19 Whitcomb Street,
London WC2
Telephone 01-839 6835

Published by Sunshine Publications Ltd

Typesetting, origination and printing by
Chesham Press, Chesham, Bucks

Distributed by S M Distribution
London SW9 01 274 8611 Telex 261643

Subscriptions
You can have *Popular Computing Weekly* sent to your home the subscription rate is £19.95 per year for addresses in the UK £37.40 overseas

How to submit articles
Articles which are submitted for publication should not be more than 1000 words long

All submissions should be typed and a double space should be left between each line

Programs should whenever possible be computer printed

At present we cannot guarantee to return every submitted article so please keep a copy

Accuracy
Popular Computing Weekly cannot accept any responsibility for any errors in programs we publish although we will always try our best to make sure programs work

This Week

Cover illustration by Stuart Hughes

Editorial

It is more than six months since the birth of *Popular Computing Weekly*. But, in that brief space of time the microcomputer market has already changed out of all recognition

The Spectrum, which arrived in April, astounded micro users with its colour, sound and 16K Ram for the ridiculously low price of £125. But it was soon followed by a range of similar micros such as the Dragon 32, Commodore 64, Colour Genie and the Lynx. The amazing has become almost commonplace

In response to the changing nature of the market *Popular Computing Weekly* is getting bigger. From November 4 we shall have 32 pages each week

This means we shall have even more programs, more news and more coverage of the minority machines. And all for the tremendously low price of 35p.

Starting next week, we shall have a Dragon page in each issue. Those Dragon owners who have been starved of software can relax at last

Spectrum, Vic, BBC and ZX81 owners will also find their needs are catered for each week

Popular Computing Weekly is going to be bigger and better than ever. Order your copy now before the rush starts

Next Week

Can you change the course of history? Find out in Guy Fawkes — a new game for 16K Spectrum.

Other features in next week's issue include a round-up of ZX81 educational software. Tony Bridge reviews the latest educational packages from ICL, Sci-Soft and others and concludes that they could do better.

Also next week, Malcolm Davison explains how to draw bar-charts to illustrate your programs

Sord M5 with 8K Rom, 4K Ram and 16K video Ram

Sord in the home

SORD Computer Systems will launch its new home micro computer in the UK by the end of November.

Called the Sord M5, it is based around the Z80A processor with 8K Rom, 4K Ram and 16K video Ram. Targetted mainly for the games market, the M5 accepts a range of plug-in Rom cartridges which provide games, languages (Basic or Pips) and utilities. Two games paddles are supplied as standard.

Video output to an ordinary tv set is in one of four modes: (a) 40 × 24 character black and white, (b) 32 × 24 character, 15 colours, (c) 64 × 48 pixel, dot programmable in 15 colours and (d) 256 × 192 pixel, only two of 15 colours in any 8 × 8 pixel (one character) area.

Up to 32 graphics shapes or sprites can be defined giving the M5 powerful animation capabilities.

Three individually programmable voices provide the tv sound output, making musical and special games effects possible.

The Sord M5 measures 10½ × 7¼ × 1½ inches and has a moving keyboard. Apart from the Rom cartridge port it has a cassette input/output, monitor video output and parallel printer output. Launched at the beginning of October in Japan there is already a library of 60 games and utilities available for the M5. Priced at around £110 in Japan, the UK price is expected to be in the region of £150.

Sord Computer Systems, founded in 1970, is Japan's fastest growing company. Sales doubled in 1981 and turn-over in 1982 is estimated at £40m. The company opened a UK office on October 1 as a prelude to launching its range of microcomputers in this country. Apart from the M5, Sord offers a range of 8- and 16-bit business systems.

Free Prestel adaptors get go-ahead

PROJECT Y, the Prestel plan to give away 100 000 adaptors, has been approved by the Board of British Telecom.

Under the scheme, customers of an as yet unnamed financial institution — believed to be a national building society — will be given free adaptors to allow their television sets to receive Prestel information.

The purpose of the package deal is to encourage more people to use British Telecom's viewdata service. At present only some 20 000 customers can access the system's 250 000 pages.

Project Y is the result of a government backed conference held in February. If the final go-ahead is given by the mystery institution, the scheme could be in full swing by January next year. It is hoped to install about 2 000 adaptors a month. Each will incorporate a full alphanumeric keyboard to allow full use of the system.

The plan will run in parallel with the Micronet 800 scheme for computer users. But where Micronet members will be able to access Prestel pages, Prestel users will not be able to call up Micronet pages.

Free Prestel adaptor

Nascom's get enhanced Basic

LUCAS Logic has produced a colour board and an enhanced Basic package for its Nascom range of microcomputers.

The Advanced Video Controller (AVC) colour board gives the Nascoms high-resolution colour graphics in three formats: a 392 x 256 mode with eight colours, a 784 x 256 mode with two colours and a combination of both modes. The AVC, which is supplied complete with a special high-resolution graphics software package, costs £185 plus VAT.

An enhanced Basic is also available on cassette for 16K Nascoms 1, 2 and 3. Enhanced Basic provides the machines with more than 75 new commands and functions including *Call, Open, Close, Chain, Create, Pop* and *Hex*. It can also cope with up to 255 files.

Lucas's Peter Horton explained that the enhanced Basic is supplied complete with an exhaustive manual which gives details of the machine-code hooks present on which you can hang your own routines. 'It gives you all the information you need to write your own Basic commands for the machine,' he said. The Nascom Enhanced Basic costs £40 plus VAT.

Hunt Inquiry report brings Cable tv nearer

CABLE television could be in operation within three years if the Government implements the recommendations of the Hunt Inquiry report published on October 12.

The main feature of the proposed guidelines of the three-man committee, headed by Lord Hunt, is the lack of restrictions. The report endorses a cable tv system with no restriction on advertising time, no vetting of material carried and no restriction on the levels of charges to customers.

Setting up a nationwide cable network would serve three main purposes, according to the report: to relay BBC, ITV, Channel 4 and radio broadcasts; to provide 'some interactive services of benefit to business and the consumer'; and to provide a large range of tv programmes of local or minority interest.

Benefit for the microcomputer user will come from the second of these three. A multi-channel cable network could give easy access to every kind of information and allow routine communications between people, computers, groups of people and groups of computers. The way is open to set up local area computer networking systems and armchair buying/selling facilities.

The extent to which cable tv will be able to fulfil these goals will depend on the precise nature of the cables used. A system using conventional coaxial cables could support about 30 channels. One based on new fibre optics cable technology would be more flexible and have many more channels. Which type of cables will be used has yet to be decided. A Department of Industry committee has been set up to advise on this question but has still to announce its findings.

Brands Hatch computer fair

SOUTH East Computers and Commodore Business Machines have combined forces to stage the South East Pet Show.

The computer fair will be held at the Kentagon, Brands Hatch from November 15-17. For more details contact Nick Manning, Haydn Manning Ltd (Tel: 0342 28358).

Beelines flight terminated

BEELINES, the Bolton based suppliers of the Beebox Vic20 expansion unit, has collapsed.

The company called in the receiver at the beginning of October. Beelines difficulties were apparently brought on by the failure of one of its sub-contractors to supply parts vital for the Beebox unit.

The collapse does not affect Beelines associated company B & B Computers, which will continue trading.

Letters

write to Letters, Popular Computing Weekly, Hobhouse Court, 19 Whitcomb Street, London WC2

Mary Goodman, Mary Goodman

Thank you for publishing my letter ('Some have many faults' August 12) Almost immediately I had a letter from Sinclair's Customer Relations department in the name of 'Mary Goodman' asking for a description of my problems When I supplied this I was asked to return the latest Ram pack to this lady at Cambridge who then refunded my £49 95

I have had to write off the cost of 'n' letters and return postages etc but I can now afford a Ram pack and a keyboard from another manufacturer

Your readers may be interested in the Customer Relations address It is Mary Goodman Customer Relations, Sinclair Research Ltd 6 Rings Parade Cambridge CB12 1SN

G D Pearce
5 Orchard Lea
Coxley Wick
Wells
Somerset

In at the deep end

I am most surprised to have twice seen incomplete and downright wrong information in what purports to be an authoritative technical information service

I refer to issues August 19 and September 9 where Ian Beardsmore in Peek & poke states that you cannot poke characters on to the ZX81 screen display

In fact this can easily be done on a 16K machine with an expanded display file by addressing relative to the system variable *D File*

The expression *Peek* 16396 + 256 * *Peek* 16397 creates *D File*, the address of the first Newline character (decimal 118) of the display file *Poking D File* + 1 with the appropriate character code fills character space (0,0) ie line 0 column 0 *D File* + 2 fills (0 1) and so on up to *D File* + 32 for (0 31) Never *Poke D File* + 33 as this contains the next Newline character Location (1,0) then must be *Poked* at *D File* + 34 (1 1) at *D File* + 35, etc

Thus one can easily calculate parameters to *Poke* any screen location, always remembering that *D File*+ any multiple of 33 is to be avoided Further as *D File* varies merely with program length it can be assigned to a variable to avoid constant use of the expression quoted previously Try the following

```
10 LET D  PEEK 16396 + 256 *
   PEEK 16397
20 INPUT N
30 POKE D + N 128
40 GOTO 20
```

More depth to your replies in future please

B Sullivan
12 Aston Road
Earlsdon
Coventry
West Midlands

Perfect partners

Although like Mr P Webb (Letters, August 19) my BBC micro was ordered last December it was very cool and delivered in perfect condition

I suspect that Mr Webb was unlucky enough to receive one of the last machines to be made before the changeover to the current switch-mode power supply This seems to have totally eliminated any problems arising from overheating

My own experience with the BBC micro is entirely favourable

J D Robinson
3 The Jinnings
Welwyn Garden City

Backdoor achievements

You may be interested to hear what I consider to be a very clever 'backdoor' achievement by Sinclair This is a redesigned circuit board for their Spectrum computer which has rendered me the 'NOT SO PROUD' owner of a DK'tronics 32K Ram module that will not fit the Spectrum

About five weeks ago I received my 16K Spectrum and noted the requirements for their upgrade to 48K whereby one had to part (after waiting 12 weeks for delivery) with the unit and £50 Having noted the DK'tronics advertisement for their 32K Ram upgrade, and realised the simplicity of fitting it in your own home for just £39 95 I jumped at the chance and ordered one

The module was fitted and worked well, until a week later, when my Spectrum failed The Ram upgrade was removed and the Spectrum returned to Sinclair Posted on August 31, I received a brand new 16K Spectrum on September 14 (only two weeks' wait, wow) only to find, horror of horrors this model had a changed circuit board design which does not allow the simple "Plug-in facility" of all Sinclair's competitors add-on Ram packs

I think this matter should be brought to light immediately to avoid a lot of people experiencing this problem I leave the matter with you and hope that my letter to DK tronics is received with sympathy

Michael Wilson
Flat 1
77 Roxborough Road
Harrow
Middlesex

Looping the loop with Spectrum

In response to Ian Logan's request for "bugs" why not try the following on your Spectrum?

```
10 FOR F = 65530 to -65540 STEP
   -1
20 PRINT F
30 NEXT F
```

This is a very interesting feature and it helps to explain the observation that *Int* −65536 gives −1 on the Spectrum (For some reason the Spectrum appears to be calculating *Int* -1×10^{-38} which is indeed −1.)

By changing line 10 (see below) it can be seen that the effect is observed only when F attempts to step to exactly −65536 It fails to do this and instead yields -1×10^{38}

```
10 FOR F = 65280 to -65800
   STEP -256
11 POKE 23692 -1 (REM
   AUTO-SCROLL)
12 PAUSE 10
NB 65280 = (65536 -256)
```

It is of interest that having looped from -1×10^{38} through to −65536 the loop stops at the correct value The explanation of this would appear to be the way in which numbers are stored by the Spectrum

Integer numbers in the range ± 65535 are stored differently from floating point numbers and numbers outside this range — see Spectrum handbook Thus the loop prints −65536 on the second time around but it is really $-(65536 + 1 \times 10^{-n})$ ie non-integer format

A number of other apparent "bugs" can also be discovered once this 'magic number' has been discovered but I will leave them for you to discover since they are only variations on a theme

PS Would someone please tell me if this bug also exists on the ZX81

M Mulheron
Dept of Metallurgy
Surrey University
Guildford

Plus que ça change . . .

Has anyone looked inside a recent Spectrum? If so, you will find that the printed circuit board has been redesigned and the 'piggy back' method of memory increase has been eliminated The additional 32K memory is now plugged into empty sockets on the main board

However something is still amiss with the ULA Two of its legs are bent up and wires run from them to another chip that has been up ended and stuck to the pcb with double-sided adhesive tape From this chip (Nandgate?) further wires run to other parts of the pcb This whole assembly is then further covered by a piece of black insulating tape

Of five Spectrums I know personally two no longer work and one gives poor colour Having seen letters regarding the Spectrum in computer magazines already I can only say 'Here we go again Uncle Clive'

I do not have a Spectrum on order and am now considering other alternative machines

D Mitchell
24 Arreton Close
Knighton
Leicester

If you have an opinion you want to express or have spotted an error that needs correcting write to Letters Popular Computing Weekly *Hobhouse Court 19 Whitcomb Street London WC2*

COVER STORY

Hallowe'en

A new game for BBC model B by Jeremy Ruston

It is Hallowe'en, and you are driving down a lonely road in the heart of Dorset. The rain beats across your face through the shattered windscreen and the lightning crashes across the sky, interfering with your Sony Walkman. Suddenly Bessy, your faithful Mini Metro, gives a disturbing hiccup and grinds to a sickening stop. The dashboard lights flicker for a few terrifying moments before they are extinguished.

With a frightened start you realise that you are alone at 11.34 pm on Hallowe'en. Terror grips you. Your eyes grow accustomed to the frighteningly dark night: the rainstorm stops, but behind you snowdrifts are now piling up with alarming speed. Darkness has turned to bright light. You realise that the only course open to you is to trek across the open countryside.

A long time later you find yourself on the brow of a hill and realise the snowdrifts have engulfed Bessy. A haunted house stands in front of you, surrounded by a forbidding looking forest.

Your only chance of survival is to enter the house and find a telephone. But watch out for the ghosts.

Now we will leave our brave traveller. The object of the game is to navigate your way through the house to the telephone. However, three ghosts are at large in the house, and they will try to stop you.

The house takes the form of several interconnected corridors. You have to travel from one side of the house to the other, but your choice of direction is limited by the position of the doors in the corridors.

A plan of the house is presented on the left-hand side of the screen. Your own position is marked in blue and that of the ghosts in white.

Two thirds of the screen is taken up with a three dimensional view of the corridor you are in. There is a time limit of one minute on the game. Elapsed time is displayed as a red column under the plan of the house.

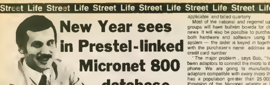
The controls are Z for left and X for right, L to go forwards and J to jump randomly to another part of the house. The jump facility may only be used once per game. You cannot go backwards.

The movement keys only work if there is a door in the appropriate place, ie you can only move forwards if there is a door in front of you. The game ends when the time is up, the ghosts have got you, or you reach the telephone.

NB The game was written for a disc based model B BBC computer, which only gives 5.75K under mode 2. Thus the game has had to be heavily compressed. If you find the game too fast, alter the value of *Del* in line 60.

```
10
20REM Haunted house
30REM (c) 1982 Jeremy Ruston
40
50ENVELOPE3,1,5,2,4,1,0,-1,1,-1,0,0,1
0,10:*FX15
60DEL=5:MODE2:VDU23;8202;0;0;0;:PROCI
NIT:REPEATPROCtime:PROCmove:PROCghosts:U
NTILTIME 6000 OR(P%DIV28)=25 ORW%:PROCen
d:*FX15,1
70END
80DEFPROCINIT LOCALT%,G%,A$,B%,J%,X%,
Y%,O,H%,Z%:*FX9

90DATA******************************
100DATA*.00 ,     000. . 00. . .000.*
110DATA*.00,.000. 000. .0000...00.*
120DATA*.....000.......000.. .000.*
130DATA* .00..00.00.00.000000. ..0.*
140DATA*.00 ..00000 .......  ..000 *
150DATA*.     0 ......0000..00......*
160DATA*00   00...00....0.  .0   0..*
170DATA*0 .00.  00 .  000..00....0*
180DATA*.  . 0   . 0. 000.   . 0..0..*
190DATA*00   00..000..000....00 .0 *
200DATA*. ...........00...0.    00 *
210DATA* 000. ..000.....0....0. ..*
220DATA*,000.  ,000...00000...00..*
230DATA*. .000..  .   0..  ...0..*
240DATA*. 000.....0000..0.......0..*
250DATA*0 .00 .00. 00...0000...00.*
260DATA*.0000 .0 .. .000.......  ..*
270DATA*.   .. .......... ..... 00*0*
280DATA*000.   000 . 000  ,000.    *
290DATA*.............................*
300DATA*  ..000...000...000...000..*
310DATA*0..000.00000..000000.000..*
320DATA*.. 000...000...000 . 000 0*
330DATA*   000   .000...000...000..*
340DATA*..................... ......*
350DATA******************************
360DIMA% 756:FORT%=0TO26:READA$:F
ORG%=1TO28:?(A%+T%*28+G%-1)=ASC(MID$(A$,
G%,1)):NEXTG%,T%:GCOLO,4:MOVE300,0:MOVE1
200,0:PLOT85,300,900:PLOT85,1200,900
370GCOL0,6:MOVE300,0:MOVE1200,0:PLOT85
,600,350:PLOT85,900,350:GCOL0,7:MOVE300,
900:MOVE1200,900:PLOT85,600,750:PLOT85,9
00,750:GCOL0,0:MOVE600,350:DRAW900,350:D
RAW900,750:DRAW600,750:DRAW600,350:MOVE3
00,900:DRAW600,750:MOVE1200,900
380DRAW900,750:B%=RND(9)*100-1000:FORT%
=0TO800STEP100:MOVE300+T%,0:DRAW600+T%DI
V3,350:FORJ%=1TO2:G=RND(1):Y%=350*G:X%=(
300+T%DIV3-T%)*G+300+T%:MOVEX%,Y%:H%=T%*
100:Z%=(300+H%DIV3-H%)*G+300+H%:DRAWZ%,Y
%:IFT%=B%ANDJ%=2 O%=X%:P%=Y%:Q%=Z%
390IFT%=B%ANDJ%=1 L%=X%:M%=Y%:N%=Z%
400NEXTJ%,T%:MOVEL%,M%:MOVEN%,M%:PLOT8
5,O%,P%:PLOT85,Q%,P%:GCOL0,0:MOVE1200,0:
DRAW900,350
410GCOL0,15:MOVE680,354:MOVE820,354:PL
OT85,680,660:PLOT85,820,660:GCOL0,12:MOV
E680,350:DRAW680,660:DRAW820,660:DRAW820
,350:MOVE800,508:PLOT1,-8,0:PLOT1,0,-8:P
LOT1,8,0:PLOT1,0,8
420MOVE700,485:MOVE740,485:PLOT85,700,
370:PLOT85,740,370:MOVE760,485:MOVE800,4
85:PLOT85,760,370:PLOT85,800,370:MOVE700
,530:MOVE740,530:PLOT85,700,640:PLOT85,7
40,640:MOVE760,530:MOVE800,530:PLOT85,76
0,640:PLOT85,800,640
430GCOL0,3:FORT%=0TO360STEP5:MOVE750,82
0:H%=100*RND(100):PLOT1,SIN(RAD(T%*135))
*H%,COS(RAD(T%*135))*H%:NEXTT%:GCOL0,0:M
OVE750,870:PLOT1,0,-30:PLOT1,-40,-40:PLO
T81,80,0:MOVE738,850:PLOT1,22,0
440GCOL0,13:MOVE400,700:MOVE400,120:PL
OT85,516,680:PLOT85,516,262:GCOL0,10:MOV
E400,120:DRAW400,700:DRAW516,680:DRAW516
,250:MOVE500,510:PLOT1,0,8:PLOT1,-8,-8:P
LOT1,0,-8:PLOT1,8,4
450MOVE420,465:MOVE450,475:PLOT85,420,
178:PLOT85,450,212:MOVE466,478:MOVE500,4
86:PLOT85,466,228:PLOT85,500,266:MOVE420
,515:MOVE450,520:PLOT85,420,664:PLOT85,4
50,660:MOVE466,524:MOVE500,534:PLOT85,46
6,660:PLOT85,500,654
460GCOL0,14:MOVE1100,700:MOVE1100,120:
PLOT85,984,68:PLOT85,984,262:GCOL0,11:M
OVE1100,120:DRAW1100,700:DRAW984,680:DRA
W984,250:MOVE1084,494:PLOT1,0,-8:PLOT1,-
8,8:PLOT1,0,8:PLOT1,8,-4
470MOVE1080,465:MOVE1050,475:PLOT85,10
80,178:PLOT85,1050,212:MOVE1034,478:MOVE
1000,488:PLOT85,1034,228:PLOT85,1000,264
:MOVE1080,515:MOVE1050,520:PLOT85,1080,6
64:PLOT85,1050,660:MOVE1034,524:MOVE1000
,534:PLOT85,1034,660:PLOT85,1000,654
480DIMU%(2):REPEATB%=RND(756)-1:UNTILA
%?B%=46:U%(0)=B%:REPEATB%=RND(756)-1:UNT
ILA%?B%=46 ANDB%<>U%(0):U%(1)=B%:REPEATB
%=RND(756)-1:UNTILA%?B%=46 ANDB%<>U%(1)
ANDB%<>U%(0):U%(2)=B%
490REPEATB%=RND(56)-1:UNTILA%?B%=46 AN
DB%<>U%(0)ANDB%<>U%(1)ANDB%<>U%(2):P%=B%
:TIME=0:GCOL0,2:MOVE0,690:DRAW224,690:DR
AW224,914:DRAW0,914:DRAW0,690:MOVE0,694
PLOT1,224,0:MOVE0,910:PLOT1,224,0:C%=TRU
E:V%=0:ENDPROC
500DEFPROCtime LOCALK%:K%=TIME DIV35:I
FK%=V%ENDPROC
510GCOL0,1:IFK%MOD2=1 MOVE 0,K%*4:PLOT
21,250,K%*4ELSEMOVE0,K%*4:PLOT21,250,K%*
4
520V%=K%:ENDPROC
530DEFPROCmove LOCALX%,Y%,B%,K%:GCOL0,
0:X%=(P%MOD28)*8:Y%=(P%DIV28)*8:PLOT69,X
%,Y%+690:PLOT69,X%,Y%+694:X%=TIME:B%=INK
EY(DEL)AND&DF:IFB%=90ANDA%?(A%+P%-1)=46 P%
=P%-1
540REPEATPROCtime:UNTILX%+DEL<TIME:IFB
%<>223SOUND17,-15,200,4
550*FX15,1
560IFB%=88AND?(A%+P%+1)=46P%=P%+1
570IFB%=76AND?(A%+P%+28)=46P%=P%+28
580IFB%=74ANDC%REPEATB%=RND(378)-1:UNT
ILA%?B%=46:P%=B%:C%=FALSE
590GCOL0,4:X%=(P%MOD28)*8:X%=(P%MOD28)
*8:Y%=(P%DIV28)*8:W%=FALSE:IFPOINT(X%,Y%
+690)<>0W%=TRUE
600PLOT69,X%,Y%+690:PLOT69,X%,Y%+694:I
FP%=OD%ENDPROC
610OD%=P%:IF?(A%+P%-1)=46VDU19,10;0;0;
19,12,RND(7);0;ELSEVDU19,10,4;0;0;19,13,
4;0;
620IF?(A%+P%+1)=46VDU19,11;0;0;19,14,R
ND(7);0;ELSEVDU19,11,4;0;0;19,14,4;0;
630IF?(A%+P%+28)=46VDU19,12;0;0;19,15,
RND(7),0;ELSEVDU19,12,4,0;0;19,15,4;0.
640ENDPROC
650DEFPROCghosts:LOCALT%,X%,Y%,L%,M%,D
%:X%=P%MOD28:Y%=P%DIV28:FORT%=0TO2:L%=U%
(T%)MOD28:M%=U%(T%)DIV28:D%=U%(T%):GCOL
,0:PLOT69,L%*8,M%*8+690:PLOT69,L%*8,M%*8
+694:F%=0:IF?(A%+D%+28)=46 F%=1
660IF?(A%+D%-28)=46F%=F%OR4
670IF?(A%+D%+1)=46F%=F%OR8
680IF?(A%+D%-1)=46F%=F%OR2
690R%=SGN(X%-L%):S%=SGN(Y%-M%):G%=0:IF
R%=1G%=8
700IFR%=-1G%=G%OR2
710IFS%=1G%=G%OR1
720IFS%=-1G%=G%OR4
730K%=F%ANDG%:IFK%=0K%=F%
740REPEATR%=RND(4)-1:UNTIL(2^R%ANDK%)
0:IFR%=0M%=M%+1
750IFR%=1L%=L%-1
760IFR%=2M%=M%-1
770IFR%=3L%=L%+1
780U%(T%)=L%+M%*28:IFU%(T%)=P%W%=TRUE
790GCOL0,7:PLOT69,L%*8,M%*8+690:PLOT69
,L%*8,M%*8+694:NEXTT%:ENDPROC
800DEFPROCend ENVELOPE 1,2,5,2,10,5,2,
5,10,1,126,-2,90,126:SOUND 17,1,200,255:
VDU17,2,"0:PRINT"The time is ";TIME DIV1
00;" and ";:IFTIME 6000PRINT"so you ran o
ut of it":ENDPROC
810IF(P%DIV28)=25PRINT"you did it! Hea
rty congrats":ENDPROC
820IFW%PRINT"the ghosts got you!"
830*FX9,3
840SOUND529,3,10,255:SOUND530,3,15,255
:SOUND531,3,20,255:*FX10,3
850GCOL0,8:DIMP%(12,1):FORT%=1TO12:P%(
T%,0)=T%*100:P%(T%,1)=-RND(30):NEXT:REPE
ATFORT%=1TO12:PLOT69,P%(T%,0),P%(T%,1):P
%(T%,1)=P%(T%,1)+4:R%=RND(3):IFR%=1P%(T%
,0)=P%(T%,0)+8
860IFR%=2P%(T%,0)=P%(T%,0)-8
870NEXT:UNTILP%(1,1)>1023
880ENDPROC
```



New Year sees in Prestel-linked Micronet 800 database

***David Kelly** talks to Bob Denton – the man behind Micronet 800.*

Two months after Bob Denton and Richard Hease got together to set up Prism Microproducts the company seems set to tie up a sizeable slice of the micro market.

Prism has been appointed the sole UK wholesaler of the Sinclair range of products. It will shortly be expanding to sell non Sinclair ZX81 software and hardware and soon software for other machines.

On January 1 1983 Prism launches Micronet 800 (*Popular Computing Weekly* September 23) an ambitious new Prestel-linked database. Micronet will, for the cost of a local telephone call, bring news, reviews and hundreds of computer programs within easy reach of your microcomputer.

The microchip first announced itself to Bob in 1972 when he worked for a cash register manufacturing company. The traumatic effect the microprocessor has had on that industry convinced him that here was something important.

Bob Denton changed industries to become marketing manager for Texas Instruments supervising the launch of the TI99/4. After that, he helped launch Mattel Intelvision, tried to save the ailing Tandata Prestel operation and most recently was Dragon's director of Sales and Marketing during the launch of the successful Dragon 32 machine.

In February this year he set up an electronics magazine on Prestel. Called *Electronics Insight* the magazine was never available as hard copy — only as pages which could be viewed on Prestel.

Then Bob met Richard Hease — Chairman of ECC and EMAP Publications. They realised that, although coming from different directions, they both wanted to set up a Prestel software network.

In June EMAP's Prestel division Telemap bought up *Electronic Insight*. These two systems, now under the control of Prism, are being expanded and enhanced and will form the basis of Micronet 800.

Ownership of Prism is split between Richard Hease its chairman and Bob its managing director. They reckon to expand the Telemap system from its current 3 000 Prestel pages up to the Micronet system which it is hoped will have 30 000 pages when it is launched in the new year. In mid-1983 the system will be gatewayed on to a GEC 4082 main-frame to provide a database of up to 150 000 pages.

Bob Denton is confident that Micronet can attract over 100 000 members in the first three years of its operation. Telemap gets 60 000 accesses to its pages each month from the 18,000 Prestel users which puts it into the top 20 information providers on Prestel. To get the hoped for number of subscribers Micronet is going to have to provide top quality information and software easily and at low cost.

"At first all we wanted to be was the catalyst in the setting up of a system like Micronet", says Bob. "In the event we ended up doing it ourselves. Micronet will do most of the things that satellite tv will do — teleshopping, armchair banking, electronic mail — at a fraction of the cost. There are now over 200,000 micros in use in the UK. Connection to Micronet by phone using a small adaptor will bring your computer to life!"

Micronet 800's Bob Denton

Micronet will cost £1 a week to members with a joining fee of around £50. For your money Micronet gives you access to the current Prestel network, news and comment, an educational software library and hundreds of programs to download listed according to machine. But Prestel subscribers will not be able to call up Micronet pages.

The cost of the system is raised from the quarterly membership fee and from advertising space sold on the pages. Much of the information and many of the programs held will be available free of charge. Authors royalties will be levied, where applicable and billed quarterly.

Most of the national and regional user groups will have bulletin boards for club news. It will also be possible to purchase both hardware and software using the system — the order is keyed in together with the purchaser's name, address and credit card number.

"The major problem", says Bob, "has been adaptors to connect the micro to the phone. We are going to manufacture adaptors compatible with every micro that has a population greater than 25 000. Provision of the Micronet adaptor is included in the joining fee.

Prism has developed three basic general-purpose hardware adaptors that will connect to a micro via an RS232 interface: a basic modem, an acoustic modem and an intelligent unit (including an auto-dial facility.

Each of the major micro manufacturers has been approached by Prism. The following is a list of machines and the expected month by which a Micronet adaptor should be available: ZX81 (March), Spectrum (March), Apple (Jan), BBC (Jan), TRS 80 (March), Commodore 3000, 4000 and 6000 (Jan), Commodore 64, 500 and 700 (March), Research Machines 380Z (Jan), Dragon 32 (June).

Adaptors are being manufactured, available in the first quarter of 1983, for Sirius, ICL, Rare, IBM, Superbrain and Dec machines. Adaptors for Sharp, Nascom, Texas Instruments, NewBrain, Atari, Lynx and Osborne are yet to be finalised.

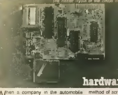
Prism plans to manufacture 100 000 adaptors, 20 000 in the first year. Bob hopes that the supply of adaptors will be a short term activity for Prism. "As Micronet takes off, more and more machines — like the Torch — will supply their own built-in adaptors."

It will cost the Micronet consortium — Prestel, EMAP, ECC and Prism — about £3m to get the scheme off the ground and a further £½m per year to keep it running. "We are probably not going to make a big profit in year one", said Bob. "What we have to do is to make it as painless as possible to join and to provide a wide range of services."

As Micronet expands so will Prism's conventional retailing outlets. After seven weeks trading the company is selling over 350 ZX81s a day. "Our privileged position with Sinclair to some extent will make Prism the arbiter of which add-ons and software are and are not bought.

"Soon Prism will be selling software for other micros. Our sales force will be marketing computer cassettes like the music business — there may even be a top 10 chart", says Bob. "In a way the two parts of Prism — telesoftware and conventional retailing — conflict. But we will be able to use Micronet as a software testing ground. We will know how often each game on Micronet is accessed. The most popular game will then be pushed in the retail outlets as 'Cassette of the Month'.

"Prism", enthused Bob, "has both ends of the market and intends to become very much a force to be reckoned with."



The neater layout of the circuit board

Reviews

Collaboration gives birth to a doubled memory

***Jeff Naylor** looks at the Timex-Sinclair 1000 and compares it with the ZX81*

When a company in the automobile industry launches a new car that is essentially the same as another car it is known as badge engineering

The Timex-Sinclair 1000, which costs $99 95 and was launched in the US last month, certainly has a different badge from the ZX81 But it also boasts twice as much memory

Timex who assemble the ZX81 and Spectrum at its plant in Dundee for Sinclair has emerged as the Timex Computer Corporation Its first micro, built under licence from Sinclair is only for sale in the US It will not be sold in the UK

The first difference to note after the badge is some of the keyboard wording There are no extra functions, but *Rubout* and *Newline* are replaced by *Delete* and *Enter* Presumably these words are thought to be more common to computer-aware Americans

All those other ZX81 features are there however — the membrane keyboard the black case the fragile jack sockets and the unplated edge connector Sorry to go on about the badge but I also noticed the raised letters ZX81 are missing Does this mean Timex has produced its own mould for the case or has the original been altered?

Turning over the Timex 1000 reveals a small switch marked ' CH2 CH3 How many devices can you plug into your television? Would it not be nice if the video cassette tv game Teletext tuner and computer(s) did not all appear on channel 36? With all the extra tv stations in America to add to the problem a channel select switch is more of a necessity than a luxury The final external difference I could spot was some white lettering referring to FCC rules and radio interference and those dreaded words NO USER SERVICEABLE PARTS INSIDE

When the case comes apart the first difference from the ZX81 is a layer of metal coating on the inside of the plastic moulding earthed via two springy metal strips protruding from the circuit board My immediate thought was that this might improve saving and loading by reducing the interference which can occur with certain juxtapositions of tv cassette and computer This was not the case however, as I soon managed to position the equipment so as to render my most reliable tape unloadable The Americans are touchy about radio interference and this neat method of screening is probably required to pass the Federal Communications Committee regulations which are referred to on the underside of the machine

Using a portable VHF radio I checked the Timex for pollution of the airwaves It was almost silent in comparison with my own Sinclair ZX81

Bird's eye view of Timex 1000

The Timex 1000 circuit board is marked "Issue 3 and ZX81 The layout is neater than earlier boards but with the exception of the Ram and a few extra components the pcb is identical to a ZX81 If you have built a Sinclair kit you will already be aware that some additional parts convert any ZX81 to American (or French) tv standards although the modulator must be of a different type

More interference suppression is provided on the 9 volt input (two coils and a capacitor) and the ear and mic sockets (capacitors) The only major physical difference between the two machines is a large 2K static Ram chip soldered firmly into the same area that is normally occupied with the ZX81 s 1K chip

What about the Rom? I loaded its contents into an array and then checked it off against a new ZX81 Rom (the one without the bug) This is the sort of boring task that computers are very good at, leaving us humans free for more rewarding tasks such as doing the washing up When we both had finished, the ZX81 confirmed an exact match between the Roms

The next problem was to get the Timex up and running Without an American television I needed to use a different modulator So I hooked up a screened lead from a point on the Timex board marked UK2 and took it to the input of my own Sinclair modulator suitably disconnected from its host computer Applying power to this botch up produced the familiar cursor but the television was struggling to lock on to only 525 lines

A study of the manual reveals a system variable called *Margin* containing the number of blank lines generated This is loaded with either 55 or 31 lines each time a tv frame is output The keyboard scanning routine also detects if a resistor called R30 is pulling pin 22 of the ULA down to earth

The practical upshot of this is if R30 is in place the computer produces 525-line pictures Remove the resistor and you have a 625-line machine! Thus, anyone who acquired a Timex 1000 could use a monitor A discarded black and white video game might provide a cheap UK modulator to turn it into a 2K ZX81

So, what difference does the extra memory make? In fact as the operating system uses upwards of 160 bytes just to work the Timex 1000 has more than twice as much space as a ZX81

When Sinclair introduced his low-priced machine the cost of memory was very high So the ZX81 Rom goes to great lengths to actually make a 1K computer work notably by setting a minimal display file if there is less than 3¼ bytes available

A full screen (768 bytes) leaves no room

Underneath the Timex 1000

for a program of any size in a 1K machine The first advantage of a 2K machine is that simple programs can use the whole screen But if you wish to write a longer listing extra room can be borrowed by keeping screen displays to the absolute minimum

There is no denying the sense of achievement in squeezing a crash proof program into a limited space but the tricks needed to do this in 1K lead to quirky inelegant programs which are difficult to understand The Timex 1000 will be a better learning machine and I believe that "lack of memory" frustration will be reached much later Many Timex owners will never buy a Ram pack but move to another computer when they need to expand

Any serious data storage is still out of the question, but software possibilities especially machine code games are greatly enhanced Many commercial prospects must exist with the size of the American market For example a game such as Artic's *Galaxians* occupies little over 3½K and includes a very elaborate title page A slimmed down version could perhaps be fitted into 2K especially as it uses less than the full screen A full feature invader game should easily fit into 2K if alternate screen lines are used

The ZX81 has already found a place in the American computer market The Timex 1000 should if pricing and marketing are right take over to great effect as a cheap consumable for Americans who are curious about computers

Improving Spectrum

Abacus Controller
Abacus Electronics 186 St Helen's Avenue, Swansea West Glamorgan
Spectrum
Price £14 95

When the Spectrum was first announced many people speculated on the possible add-ons that would be offered for it, given that the bottom had fallen out of the market for keyboards/add on Rams and high resolution graphics Of all the Spectrum hardware items I have reviewed this one most impressed me and is the only one I shall always use

There are two serious design faults with the Spectrum One is the need to unplug the cassette plug not being used — which is ludicrous on a machine of this calibre — and the other is the silent beeper This simple device solves both those problems, and makes the Spectrum a much nicer machine in the process

The Abacus Controller is fitted with one socket, one switch and five flying leads To use it, you unplug the power socket from the back of the Spectrum and push it into a similar socket on the Controller Sinclair's cassette leads can be discarded, as two Controller leads go into the Spectrum's cassette sockets, while another two connect to similar sockets on the cassette recorder Once the power lead from the Controller is plugged into the Spectrum, you are ready for business

The Controller has a three position rotary switch, labelled *Load*, *Save* and *Amp* The switch is set to the first two positions when *Loading* and *Saving* The *Amp* position is used when an amplifier is required for the Beeper

The volume control for the beeper is accessed with a small screwdriver through a hole in the case — a serious inconvenience

The device also amplifies the cassette signals so you have to adjust the volume levels on your cassette recorder

Summary

All Spectrum owners should seriously consider buying an Abacus Controller or an equivalent but only when Abacus has been lobbied to supply longer leads **JR**

Magic from the little blackbox

Beebstick Micrex 54 Linley Road Alsager Stoke-on-Trent Staffordshire Tel Alsager 77270
Price £29 95 inc VAT and packaging

The Beebstick comes in a strong cardboard box well packed with polystyrene for protection It also has a cassette with various demonstration programs suitable for a model A machine fitted with an analogue to digital converter, or a model B

The instructions on the Beebstick's use are very simple First plug in the joystick to the D shaped, multipin socket at the back Then read the instructions on how to adjust your programs to accommodate the Beebstick by using the BBC Basic variables *Adval0* to *Adval2* *Adval0* returns a 2 if the fire button has been pressed *Adval1* returns a value between 0 and 65535 for the horizontal value and *Adval2* the same for the vertical value

The range of numbers returned by the Beebstick are so great that they need to be scaled down to allow the user to move

Beebstick

from one dot to another But this is clearly explained in the instructions

The demonstration programs on the accompanying cassette are simple but show the usefulness of the Beebstick The *Sketch* program is my favourite as I was able to amaze my six year-old by drawing her name on the screen in normal handwriting

The Beebstick itself is very easy to use It is very similar to those joysticks used for controlling model planes

The stick is spring loaded into the middle position and the fire button on the top left hand corner is in easy reach The black box is 6 x 3 x 2 inches The ribbon cable provided is a generous two feet six inches and consists of a 16-way ribbon cable with D type plug on the end

Conclusion

This one of the reviews I enjoyed doing as the device is so simple to understand and use It is robust, useful, and reasonably priced Micrex would like to hear of its use for handicapped people **SA**

ZX81 Graphics Rom

4K Graphics Rom
Kayde Electronic Systems Ltd, The Conge Great Yarmouth Norfolk Tel 0493-55253
ZX81
Price £29 95 inc VAT

The Kayde 4K graphics Rom gives the ZX81 a choice of eight different character sets selectable by *Usr* calls Kayde has also taken the sensible course of issuing software to take advantage of the board

Fitting the board is not simple You must unplug the Rom from the Sinclair main board, plug it into the graphics board and solder four wires to the main pcb Not a job for the beginner but someone at your local user club would probably do it for you The instructions are faultless

Once installed a *Rand Usr X* will select one of the character sets Character set number 1 is the standard Sinclair set, number 2 contains various faces and musical symbols (but no numbers or letters) while number 3 contains some Pacman symbols digits and the playing card symbols Number 4 gives you upper and lower case letters and punctuation symbols but no digits number 5 gives you an assortment of graphic symbols letters and digits and number 6 contains more Pacman symbols and letters and digits Number 7 gives you all the asteroids characters and digits and finally number 8 gives you digits and an assortment of games symbols

Only one of the sets can be on the screen at the same time giving a funny appearance to program listings! I was sorry to see that none of the sets made any use of the *Plot* statement — it would have been nice if the *Plot* command would give some sort of recognisable result

Also available is Kayde's version of *Pacman* *Peckman* (£5 95) which takes advantage of the facilities of the graphics Rom It was a pity I reviewed Acornsoft's *Pacman* for the BBC Computer on the same day, but Kayde's version is still excellent, given the limitations of the ZX81 even with the graphics Rom I can't wait to see Kayde's version for the Spectrum

Summary

These two products improve the ZX81 beyond all recognition for games playing If that is your forte you should give serious thought to both products **JR**

Open Forum

Open Forum is for you to publish your programs and ideas. It is important that your programs are bug free before you send them in. We cannot test all of them. Contributions should be sent to: Popular Computing Weekly, Hobhouse Court, 19 Whitcomb Street, London WC2H 7HF

How to contribute

Each week the editor goes through all the programs that you send to Open Forum in order to find the Program of the Week.

The author of that program will qualify for DOUBLE the usual fee we pay for published programs. (The usual fee is £5.)

Presentation hints

Programs which are most likely to be considered for the Program of the Week will be computer printed and accompanied by a cassette.

The program will be well documented, the documentation being typed with a double spacing between each line.

The documentation should start with a general description of the program and then give some detail of how the program has been constructed and of its special features.

Listings taken from a ZX Printer should be cut into convenient lengths and carefully stuck down on to white paper, avoiding any creasing.

Please enclose a stamped, self-addressed envelope.

Cypher

on ZX81

This program is a computerised version of an extremely secure cypher system used by a 16th Century French cryptographer Blaise de Vigenere, when travelling widely in Europe on diplomatic service.

Unlike a monoalphabetic substitution cypher, in which each letter of the message is always represented by the same letter — for example "a" equals "f", "g" equals "l" etc — in a Vigenère cypher each letter is represented by a letter which is dependent on a keyword known only to the originator of the message and the addressee.

This keyword determines which column of Vigenère table (see Fig 1) will be used when transposing the letters of the message, and thereby provides an extremely secure means of passing a message to a friend, or of keeping records secret from inquisitive snoopers.

Program notes (fig 2)

- 40 to 130 set up the Vigenère table
- 140 to 250 accept the message to be encyphered or decyphered
- 260 to 310 determine whether to encypher or decypher
- 320 to 430 constitute the encypher routine
- 440 to 550 constitute the decypher routine

To use the program

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
BCDEFGHIJKLMNOPQRSTUVWXYZA
CDEFGHIJKLMNOPQRSTUVWXYZAB
DEFGHIJKLMNOPQRSTUVWXYZABC
EFGHIJKLMNOPQRSTUVWXYZABCD
FGHIJKLMNOPQRSTUVWXYZABCDE
GHIJKLMNOPQRSTUVWXYZABCDEF
HIJKLMNOPQRSTUVWXYZABCDEFG
IJKLMNOPQRSTUVWXYZABCDEFGH
JKLMNOPQRSTUVWXYZABCDEFGHI
KLMNOPQRSTUVWXYZABCDEFGHIJ
LMNOPQRSTUVWXYZABCDEFGHIJK
MNOPQRSTUVWXYZABCDEFGHIJKL
NOPQRSTUVWXYZABCDEFGHIJKLM
OPQRSTUVWXYZABCDEFGHIJKLMN
PQRSTUVWXYZABCDEFGHIJKLMNO
QRSTUVWXYZABCDEFGHIJKLMNOP
RSTUVWXYZABCDEFGHIJKLMNOPQ
STUVWXYZABCDEFGHIJKLMNOPQR
TUVWXYZABCDEFGHIJKLMNOPQRS
UVWXYZABCDEFGHIJKLMNOPQRST
VWXYZABCDEFGHIJKLMNOPQRSTU
WXYZABCDEFGHIJKLMNOPQRSTUV
XYZABCDEFGHIJKLMNOPQRSTUVW
YZABCDEFGHIJKLMNOPQRSTUVWX
ZABCDEFGHIJKLMNOPQRSTUVWXY
```

Fig 1 De Vigenere Table

```
  10 REM POLYALPHABETIC SUBSTITU
TION CYPHER
  20 REM AUTHOR - A.D.CROY
  30 REM CREATE SUBSTITUTION TAB
LE
  40 FAST
  50 DIM A(26,26)
  60 LET J=-1
  70 FOR I=1 TO 26
  80 LET J=J+1
  90 FOR K=1 TO 26
 100 LET A(I,K)=K+J
 110 IF A(I,K)>26 THEN LET A(I,K
)=A(I,K)-26
 120 NEXT K
 130 NEXT I
 140 REM START MESSAGE ROUTINE
 150 PRINT
 160 PRINT
 170 SLOW
 180 PRINT "    ENTER KEYWORD"
 190 INPUT K$
 200 PRINT K$
 210 PRINT
 220 PRINT "    ENTER MESSAGE"
 230 INPUT M$
 240 PRINT M$
 250 PRINT
 260 PRINT "    ENCYPHER/DECYPHER?
(ENTER E OR D)"
 270 INPUT Q$
 280 PRINT
 290 IF Q$="E" THEN GOTO 320
 300 IF Q$="D" THEN GOTO 440
 310 IF Q$<>"E" OR Q$<>"D" THEN
GOTO 260
 320 REM ENCYPHER ROUTINE
 330 FAST
 340 PRINT "    ENCYPHERED MESSAGE
READS -"
 350 PRINT
 360 LET S=0
 370 LET T=0
 380 LET S=S+1
 390 IF S>LEN K$ THEN LET S=S-LE
N K$
 400 LET T=T+1
 410 PRINT CHR$ (37+A(CODE K$(S)
-37,CODE M$(T)-37));
 420 IF T=LEN M$ THEN GOTO 140
 430 GOTO 380
 440 REM DECYPHER ROUTINE
 450 FAST
 460 LET Z=0
 470 PRINT "    DECYPHERED MESSAGE
READS -"
 480 PRINT
 490 FOR W=1 TO LEN M$
 500 LET Z=Z+1
 510 IF Z>LEN K$ THEN LET Z=Z-LE
N K$
 520 FOR X=1 TO 26
 530 IF A(CODE K$(Z)-37,X)=CODE
M$(W)-37 THEN PRINT CHR$ (X+37);
 540 NEXT X
 550 NEXT W
 560 GOTO 140
```

Fig 2 Polyalphabetic Substitution Cypher Program

(1) You and your correspondent agree on a keyword or phrase, for example NOWISTHETIME (Note: No spaces or punctuation marks)
(2) Feed in the program
(3) Follow the instructions in the program
(4) Copy the result of the encypherment or decypherment

Examples of messages are shown in Figs 3 & 4

```
ENTER KEYWORD
NOWISTHETIME

ENTER MESSAGE
HAVEFOUNDTHEGOLDBRINGRICKSANDSHO
VELSJOE

ENCYPHER/DECYPHER?(ENTER E OR D)

ENCYPHERED MESSAGE READS -

JORMXHMRWETITCHLTKPRZXUSXQUVULOS
JMXWUCR
```

Fig 3 Encypherment

```
  ENTER KEYWORD
NOWISTHETIME

  ENTER MESSAGE
JORMXHMRWETITCHLTKPRZXUSXQUVULOS
JMXWUCR

  ENCYPHER/DECYPHER?(ENTER E OR
D)

  DECYPHERED MESSAGE READS -

HAVEFOUNDTHEGOLDBRINGRICKSANDSHO
VELSJOE
  ENTER KEYWORD
```

Fig 4 Decypherment

Cypher
by Alan Croy

3D Graphics

on ZX81

Until a three dimensional television is available true 3D plotting is impossible. By using the principles involved in simple geometry, however, movement in three dimensions can be simulated. The functions involved are scaling, perspective and rotation, all of which can be carried out using some very simple equations and functions such as Sin and Cos.

The example program given at the end of the article is written for a ZX81. It may be used on any computer with *Plot* or some such equivalent statement such as

Set A number of variables are used in the program which will need special alteration for other machines they are

MX and MY the maximum x- and y-coordinates

CX and CY, the centre about which the shapes are plotted For convenience these are set to the middle of the screen, at ½MX and ½MY

X(n) Y(n) and Z(n), the X, Y and Z coordinates of the point Only the X and Y coordinates are plotted, the Z is used to calculate perspective

Plot works with two parameters, X and Y These allow two dimensions The Z axis can be imagined as being at right angles to the other axes, coming directly out from the centre of the screen A positive Z coordinate denotes a position in front of the tv a negative Z is behind the picture The greater the Z coordinate the closer the image will appear to be

Most computers have the zero points on the axes at a corner of the screen which is not very convenient for functions Therefore a false centre must be set up usually in the centre of the image In the illustration program CX and CY denote the artificial centre To account for this a point X Y where X and Y are relative to CX and CY would be plotted as

PLOT CX+X CY+Y

When a point is plotted in the program the array coordinates are not used in the *Plot* statement X and Y are used instead, X() Y() and Z() are used to calculate the final position If a number of complex shapes must be moved around it is simpler to have a separate centre for each shape

This is the process which determines whether a point lies within the legal limits for the *Plot* statement A point is illegal if, with CX and CY added the point is greater than the MX and MY limits or less than zero *Plotting* outside these limits will normally cause an error

The Z coordinate is used as an offset to calculate for perspective If Z() is positive then the point is further away from the centre than if Z would be zero F is used to multiply a point to calculate the offset I used 2 if Z was positive 0.5 if it was negative or 1 if it was zero You should change this according to taste

If the entire shape is to be moved by a certain amount without the relationship of individual points being changed then it is simpler to alter the centre about which the shape is plotted If rotation, stretching or shearing is needed then the following simple mathematical functions should be used

Rotation

If a shape needs rotating about any axis by N degrees or radians depending on your computer the following functions need to be used

Rotation about the Z axis
X=X•COS(N)−Y•SIN(N)
Y=Y•SIN(N)+X•COS(N)
Z=Z

Rotation about the X axis
Y=Y•COS(N) Z•SIN(N)
Z=Z•SIN(N)+Y•COS(N)
X=X

Rotation about the Y axis
X=X•COS(N)−Z•SIN(N)
Z=Z•SIN(N)+X•COS(N)
Y=Y

Objects may be stretched in any direction by increasing the X Y or Z coordinates by any amount

Shearing involves adding the Y coordinate to the X or Z coordinate and stretching by 45°

```
   10 REM 50 50HO,R.ESMOND 27 9.8
2
   11 REM TO SAVE SPACE ONLY THE
POINTS ARE PLOTTED,THEY ARE NOT
JOINED
   20 REM ZX81 VERSION
   30 LET MX=63
   40 LET MY=43
   50 LET CX=MX/2
   60 LET CY=MY/2
   70 DIM X(8)
   80 DIM Y(8)
   90 DIM Z(8)
  100 LET M=8
  110 LET RL=1100
  140 LET X(1)=4
  150 LET X(2)=4
  160 LET X(3)=-4
  170 LET X(4)=-4
  180 LET Y(1)=4
  190 LET Y(2)=-4
  200 LET Y(3)=-4
  210 LET Y(4)=4
  220 FOR F=1 TO 4
  230 LET X(4+F)=X(F)
  240 LET Y(4+F)=Y(F)
  250 LET Z(F)=4
  260 LET Z(F+4)=-4
  270 NEXT F
 1000 REM PLOT FRONT VIEW
 1010 CLS
 1011 FOR F=1 TO 8
 1015 LET SF=1
 1016 REM SCALE FOR PERSPECTIVE
 1020 IF SGN Z(F)=1 THEN LET SF=1
.5
 1030 IF SGN Z(F)=-1 THEN LET SF=
.5
 1040 REM CALCULATE TRUE POSN
 1045 LET X=X(F)*SF+CX
 1050 LET Y=Y(F)*SF+CY
 1060 REM CLIP
 1060 IF (X>MX OR X<0) OR (Y>MY O
R Y<0) THEN GOTO 1090
 1070 PLOT X,Y
 1080 NEXT F
 1090 GOTO RL
 1100 REM STRETCH IT
 1110 FOR F=1 TO 8
 1120 LET X(F)=X(F)+(4*SGN X(F))
 1130 NEXT F
 1140 LET RL=1150
 1141 GOTO 1010
 1150 REM NOW IN Y AND Z AXES
 1160 FOR F=1 TO 8
 1170 LET Y(F)=Y(F)+(4*SGN Y(F))
 1180 LET Z(F)=Z(F)+(4*SGN Z(F))
 1190 NEXT F
 1200 LET RL=1220
 1210 GOTO 1010
 1220 REM ROTATE IT BY 45 DEGS=(4
5/180*PI)RADS
 1221 REM ABOUT Z AXIS
 1230 LET RO=(45/180*PI)
 1240 FOR F=1 TO 8
 1250 LET Y=Y(F)*SIN RO+X(F)*COS
RO
 1260 LET X=X(F)*COS RO-Y(F)*SIN
RO
 1270 LET X(F)=X
 1280 LET Y(F)=Y
 1290 NEXT F
 1300 LET RL=9999
 1310 GOTO 1010

 THE VARIABLE RL IS USED WITH GO
TO RL INSTEAD OF GOSUB
```

3D Graphics
by Andrew Esmond

Donkey

on ZX81

Donkey Kong is an original game of rescuing a 'damsel in distress' from the clutches of King Kong

You control a little man and you have to climb up ladders and along girders to rescue the princess But unfortunately all sorts of things seem to get in your way including barrels which roll along the girders There are several holes in each girder which you must leap over otherwise you fall down with stunning effects

If you manage to reach the princess a love heart appears but is cracked as Kong grabs the princess back into his clutches Your score is displayed when you reach the princess and varies depending on how quickly you reach the princess Don't be scared by the beating of Kong's chest otherwise you may lose your concentration

Here is a tip on jumping over barrels Wait till the barrel is directly in front or behind you before you attempt to leap over it Be quick on the button as the next barrel rolls towards you Lastly make sure you do not hang around near the end of the screen, otherwise you may be hit as the barrels change direction

Controls — '5'-LEFT, '8' RIGHT '7'-CLIMB '1'-JUMP RIGHT, 'T'-JUMP LEFT

The game uses a 24 line screen which is done by having a POKE 16418 0 in the program

Main Variables —
S SCORE G BONUS C/D/F/E BARRELS A/B YOUR POSITION
U$/B$/AG-check to see what you are hitting
350 510 MAIN GAME ROUTINE 6000 LADDER CLIMBING
7000 8020 JUMPING ROUTINES
9000 9070 GAME OVER ROUTINE 9080 9130 WINNING ROUTINE

```
   20 LET L=1
   30 LET D=21
   40 LET F=-1
   50 LET E=1
   60 LET R=22
   70 LET B=4
  200 PRINT AT 7,0, ...
  250 FOR N=1 TO 4
  270 NEXT N
  280 PRINT AT 8,7 ...
  290 PRINT AT 5,7 ...
  300 PRINT AT 4,7 ...
  310 PRINT AT 3,5 ...
  320 PRINT AT 2,5 ...
  330 PRINT AT 20,0; ... AT 21,0
```



```
T 14 0,  0 0 0  AT 18,C,  0
0 0 ,AT 22,0,  0 0 0
351 PRINT AT A,B
352 LET A$=CHR$ PEEK (PEEK 1639
5+256*PEEK 16396)
353 PRINT AT A,B,
354 IF A$= 0 THEN GOTO 9000
360 LET C=C+E
370 LET D=D+F
375 PRINT AT 0,10, = AT 4 10
,AT 1,11, HELP"
376 PRINT AT A+1,B,
377 LET B$=CHR$ PEEK (PEEK 1639
6+256*PEEK 16396)
378 IF B$=   THEN GOTO 600
380 IF C=21 THEN LET E=-1
390 IF C=2 THEN LET E=1
400 IF D=2 THEN LET F=1
410 IF D=21 THEN LET F=-1
411 LET G=G-1
412 IF A>0 AND G=7 THEN GOTO 99
00
480 PRINT AT A,B A$
470 IF INKEY$= 7  AND A$= N  TH
EN GOSUB 5000
480 LET B=B+(INKEY$= 8  AND B<3
0) (INKEY$= 5  AND B>0)
490 IF INKEY$= I  THEN GOSUB 70
00
500 IF INKEY$= T  THEN GOSUB 80
00
505 PRINT AT 0,10, =  AT 4,10
,AT 1,11,
510 GOTO 350
600 PRINT AT A,B,   ,AT A+1,B,

610 PRINT AT A+2,B,
620 LET B$=CHR$ PEEK (PEEK 1639
6+256*PEEK 16396)
630 IF B$<>   THEN LET A=A+1
631 IF B$<>   THEN GOTO 9000
632 LET A=A+1
650 GOTO 600
5000 FOR N=A TO A-3 STEP -1
5001 PRINT AT N,B,
5002 FOR J=1 TO 3
5003 NEXT J
5004 PRINT AT N B, H
5005 NEXT N
5006 PRINT AT N,B,   AT N B,
5007 LET A=A-4
5008 RETURN
7000 PRINT AT A,B,   AT A-1 B,
  AT A-1 B    AT A-2 B    AT
A-2,B, AT A-2 B+1,  AT A-2
B+1,  AT A-2 B+2  AT A-2 B+
2,  AT A-2 B+2   AT A-1,B+2
  AT A 1,B+2,
7001 PRINT AT A B+2,
7002 LET U$=CHR$ PEEK (PEEK 1639
6+256*PEEK 16396)
7003 IF U$= 0  THEN GOTO 9000
7004 PRINT AT A B+2    AT A,B+2

7010 LET B=B+2
7020 RETURN
8000 PRINT AT A,B,    AT A-1 B,
  AT A-1 B,   AT A-2,B   AT
A-2 B    AT A-2,B-1,   AT A-2
B-1    AT A-2,B-2,   AT A-2,B-
2    AT A-1,B-2    AT A-1,B-2

8001 PRINT AT A,B-2,
8002 LET U$=CHR$ PEEK (PEEK 1639
6+256*PEEK 16396)
8003 IF U$= 0  THEN GOTO 9000
8004 PRINT AT A,B-2,    AT A,B-2

8010 LET B=B-2
8020 RETURN
9000 FOR N=1 TO 30
9010 PRINT AT A,B,  ,AT A B,
,AT A B    AT A B
9030 NEXT N
9040 PRINT AT 10 10  GAME-OVER
9050 PAUSE 200
9060 CLS
9070 GOTO 1
9900 PRINT AT 2 25      AT 3,25

9901 LET S=S+G
9990 FOR N=1 TO 50
9991 NEXT N
9992 PRINT AT 2,25    ,AT 3,25

9993 PRINT AT 1 0  SCORE  ,S
9994 PRINT AT 3         ,AT 2
     ,AT 1,7,
9995 PRINT AT 4,16      AT 3,1
5,   ,0 HELP
9100 FOR U=1 TO 50
9120 NEXT U
9130 CLS
9130 GOTO 10
```

Donkey
by Simon Cox

Tank Battle

on Vic 20

The program runs on an unexpanded Commodore Vic20 with joystick. It uses hi-resolution graphics. The game is called Tank Battle and is for two players. The object is to shoot the opposing player's tank. One player, who starts at the bottom right of the screen, uses the joystick. The other player uses the keyboard.

Full instructions are included in the program. It is important that no buttons on the cassette deck are pushed down when the program is run because it interferes with the joystick *Peeks*.

Program notes

4-90	Set up screen
110	Set variables of tanks
120-140	*Peeks* for joystick and keys.
150-190	Move and check fire button for key player
200-240	Move and check fire button for joystick player
250-300	Fire and bullet peeks for joystick player
310-335	Fire and bullet peeks for key player
400-417	Sub-routines for direction of players
1000-1490	Instructions
1500-1600	Hi-res
2000-2510	End program routine and ask if another game is required

```
1 REMTANK BATTLE BY PHILIP JONES
4 PRINT "HOW MANY OBSTACLES 1-MANY TO 5-FEW"
5 INPUTT:IFT<1ORT>5THEN5
6 T=T+1
10 PRINT "":POKE36879,27:PRINT "INSTRUCTOINS ?
(Y/N)"
20 GETA$:IFA$=""THEN20
30 IFA$="Y"THENGOSUB1000:PRINT ""
40 GOSUB1500
45 PRINT ""
46 GOTO90
50 FORA=7680TO7701:POKEA,9:NEXT
60 FORA=7701TO8185STEP22:POKEA,9:NEXT
70 FORA=8185TO8164STEP-1:POKEA,9:NEXT
80 FORA=8164TO7680STEP-22:POKEA,9:NEXT:GOTO110
90 FORA=7703TO8162:B=INT(RND(1)*T)
100 IFB=0THENPOKEA,9
105 NEXT:GOTO50
110 A=4:B=0:C=7703:D=8162
120 POKEC,A:POKED,0
130 E=PEEK(197)
140 F=PEEK(37151):POKE37154,127:G=PEEK(37152):
POKE37154,255
150 IFE=42THENA=A-1:IFA=-1THENA=7
160 IFE=3THENA=A+1:IFA=8THENA=0
165 IFA=1THEN17
170 IFE=33THENH=1:GOSUB400:C=J:I=C+J
175 POKEC,A
180 IFE=1THENGOSUB400:C=C+1
185 GOSUB500
190 POKEC,A:POKEC+30720,
200 IFF=110THENB=B-1:IFB=-1THENB=7
210 IFF=126ANDG=119THENB=B+1:IFB=8THENB=0
215 IFM=1THEN2
220 IFF=94THENM=1:GOSUB410:L=D+F
225 POKED,B
230 IFF=122ANDG=247THENGOSUB410:D=D+L
235 GOSUB600
240 POKED,B:POKED+30720,7
250 IFN=1THEN310
254 IFPEEK(I)=8THENN=0:GOTO10
255 POKEI,
260 O=O+L
264 IFPEEK(O)=2THENN=0:GOTO1
265 IFPEEK(O)=8THENN=0:GOTO310
270 IFPEEK(O)=THENPOKEO,2:N=0:GOTO10
280 IFPEEK(O)=10THENPOKEO,H=0,H=0:GOTO10
290 IFPEEK(O)=8THEN
300 POKEO,10
310 IFA=1THEN
311 IFPEEK(I)=THENH=0:GOTO10
314 POKEI,
315 I=I+J
320 IFPEEK(I)=THENH=0:GOTO340
325 IFPEEK(I)=9THENPOKEI,H=0.GOTO340
330 IFPEEK(I)=6THEN2500
335 POKEI,10
340 GOTO140
400 IFA=0THENJ=-22:RETURN
401 IFA=1THENJ=-21:RETURN
402 IFA=2THENJ=1:RETURN
403 IFA=3THENJ=23:RETURN
404 IFA=4THENJ=22:RETURN
405 IFA=5THENJ=21:RETURN
406 IFA=6THENJ=-1:RETURN
407 IFA=7THENJ=-23:RETURN
410 IFB=0THENL=-22:RETURN
411 IFB=1THENL=-21:RETURN
412 IFB=2THENL=1:RETURN
413 IFB=3THENL=23:RETURN
414 IFB=4THENL=22:RETURN
415 IFB=5THENL=21:RETURN
416 IFB=6THENL=-1:RETURN
417 IFB=7THENL=-23
500 IFPEEK(C)<>32THENC=C-J
510 RETURN
600 IFPEEK(D)<>32THEND=D-L
610 RETURN
1000 PRINT "INSTRUCTIONS"
1010 PRINT "CONTROLLER OF YELLOW TANK (BOTTOM
RIGHT) USES JOYSTICK"
1020 PRINT "STICK LEFT -TURN ANTI CLOCKWISE"
1030 PRINT "STICK RIGHT -TURNS CLOCKWISE"
1040 PRINT "STICK FOREWARD THRUST"
1050 PRINT "BUTTON - FIRE"
1060 PRINT "HIT A KEY"
1070 POKE198,0:WAIT198,1
1080 PRINT "OTHER PLAYER USES KEYS"
1090 PRINT "Q AND B TURN"
1100 PRINT "U THRUST"
1110 PRINT "F FIRE"
1120 PRINT "HIT A KEY"
1130 POKE198,0:WAIT198,1
1140 PRINT "":RETURN
1499 RETURN
1500 REM
1510 POKE
1520 POKE
```

PROGRAM OF THE WEEK

```
[illegible]
```

Tank Battle
by Philip Jones

Pacman

on Vic20

Munch your way round the maze eating as many dots and power pills as possible. Watch out for the four ghosts that haunt the maze. If they catch you, you are dead. When you eat a power pill the ghosts turn yellow for about ten seconds. During this time you can chase the ghosts. But watch out, they have a nasty habit of turning black just as you are about to eat them.

Controls

3 Moves left
4 Moves right
6 Moves up
7 Moves down

Scoring

Dots=10 points
Power pills (diamonds)=100 points
Ghosts (when yellow)=500 or 100 points

The author's hi score is 20160. Can you beat it?

Program notes

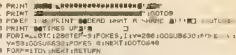
0 to 8 define characters.
9 to 11 main variables in the program
15 to 25 control the ghosts
30 check for time up
40 has ghost hit you?
50-54 scan keyboard and tell which key was pressed
55-60 move Pacman and check what it has hit
62-63 see if Pacman uses entrances at side of maze
100 print score
101 timer to change ghost from yellow to black
500-521 draw maze
600-680 end of game

```
0 POKE51,0:POKE52,28:POKE55,0:POKE56,28:CLR:X=7168:XX=32768
1 IFX=7680THENX=7432:GOTO3
2 POKEX,PEEK(XX):X=X+1:XX=XX+1:GOTO1
3 READXX:IFXX=-1THEN5
4 POKEX,XX:X=X+1:GOTO3
5 V=36869:POKEV,255:POKEV+9,15:GOTO500
6 DATA16,40,84,186,84,40,16,0,56,124,214,214,254,254,218,
  146,0,0,0,24,24,0,0,0,60,126,56
7 DATA31,31,63,126,60,56,126,236,248,248,252,126,60,60,
  126,255,251,255,231,66,0,0,66,231
8 DATA255,223,255,126,60,-1
9 O(1)=7863:O(2)=7869:O(3)=7995:O(4)=8001:O=34:PM=37:
  D(1)=1:D(2)=-1:D(3)=-22:D(4)=22
10 DM=33:DT=35:S=V+7:DEFFNR(X)=INT(RND(1)*X)+1:POKE198,
  0:P=8130:CO=30720:D=1:DIMY(4)
11 TI$="000000":SC=0:OC=0:S=V+7:POKEV+10,25
15 FORI=1TO4
16 Y(I)=D(FNR(4)):IFPEEK(O(I)+Y(I))=230THEN16
17 IFO(I)+Y(I)=7922ORO(I)+Y(I)=7942THEN16
18 IFPEEK(O(I)+Y(I))=PMANDOC=0THENI=5:GOTO600
19 IFPEEK(O(I)+Y(I))=PMANDOC=7THENSC=SC+500:POKES,200
24 POKEO(I),DT:POKEO(I)+CO,3:IFFNR(60)=1THENPOKEO(I),DM:
  POKEO(I)+CO,4
25 O(I)=O(I)+Y(I):POKEO(I),O:POKEO(I)+CO,OC:POKES,0:NEXT
30 IFTI$>="000230"THEN685
40 IF(O(1)=PORO(2)=PORO(3)=PORO(4)=P)ANDOC=0THEN600
49 POKEP,PM:POKEP+CO,2:IFQ=1THENQ=0:GOTO55
50 J=PEEK(203):IFJ=64THEN55
51 IFJ=1THENZ=D(2):PM=36
52 IFJ=57THENZ=D(1):PM=37
53 IFJ=58THENZ=D(3):PM=39
54 IFJ=3THENZ=D(4):PM=38
55 IFPEEK(P+Z)=230THEN100
56 IFPEEK(P+Z)=DTTHENSC=SC+10:POKES,230
57 IFPEEK(P+Z)=DMANDOC=0THEN600
58 IFPEEK(P+Z)=DMANDOC=7THENSC=SC+500:POKES,180
59 IFPEEK(P+Z)=DMTHENOC=7:U=0:SC=SC+100:POKES,240
60 POKEP,32:P=P+Z:POKEP,PM:POKEP+CO,2
62 IFP=7922ANDZ=-1THENPOKEP,32:P=7942:Q=1
63 IFP=7942ANDZ=1THENPOKEP,32:P=7922:Q=1
99 POKES,0
100 PRINT"SCORE"SC""
101 U=U+1:IFU=15THENOC=0
499 GOTO15
500 PRINT"      MINI PAC MAN
501 PRINT"
502 PRINT" ..................
503 PRINT"
504 PRINT"
505 PRINT"
506 PRINT"
507 PRINT"
508 PRINT"
```

turn to next page

from page 17

```
509 PRINT [illegible]
510 PRINT [illegible]
511 PRINT [illegible]
512 PRINT [illegible]
513 PRINT [illegible]
514 PRINT [illegible]
515 PRINT [illegible]
516 PRINT [illegible]
517 PRINT [illegible]
518 PRINT [illegible]
519 PRINT [illegible]
```



```
520 PRINT [illegible]
521 PRINT [illegible]";GOTO9
600 POKE 1 8 PRINT"[illegible]DEAD WHAT A SHAME [illegible]
605 PRINT"[illegible]TIMES UP[illegible]
610 FORI=[illegible]TO[illegible]:POKES,I:V=200:GOSUB630:[illegible]
    V=50:GOSUB630:POKES 0:NEXT:GOTO640
630 FORP=1TO[illegible]:NEXT:RETURN
640 PRINT"[illegible]YOUR SCORE WAS";SC:POKEV,240
660 IFSC<5000THENPRINT"GIVE UP!":RUN
670 IFSC<10000THENPRINT"NOT BAD.":RUN
680 PRINT"THE GHOSTS HAVE FORMEDAN ANTI PAC-MAN UNION
RUN
```

Pacman
by Ian Henderson

Skiing

on Spectrum

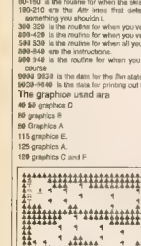
Skiing involves a slalom skier manoeuvring down a course to the finish post. The graphics are printed on to the screen by a *Read* and *Data* statement. *Bin* statements allow the user to use the hi-res graphics facility. I have also written a full set of instructions into the program.

Program notes

5-40 set up the hi-res graphics
40-80 print out the board
80-160 is the routine for when the skier is moving
180-210 are the *Attr* lines that detect if you've hit something you shouldn't.
300-320 is the routine for when you've hit a flag
400-420 is the routine for when you've hit a tree
500-530 is the routine for when all your skiers die
800-840 are the instructions.
900-940 is the routine for when you've finished the course
9000-9030 is the data for the *Bin* statements
9030-9040 is the data for printing out the course

The graphics used are

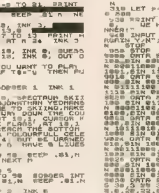
40 50 graphics D
80 graphics B
80 Graphics A
115 graphics E.
125 graphics A.
120 graphics C and F



```
SPECTRUM SKIING BY
©JONATHAN YEOMANS

WELCOME TO SKIING.MAKE YOUR
WAY DOWN THE COURSE USING THE
CURSOR KEYS TO DODGE THE FLAGS
AND FOREST TO REACH THE BOTTOM
AND A COLOURFULL CELEBRATION

BUT BE WARNED YOU ONLY
HAVE 5 LIVES
```



Skiing
by Jonathon Yeomans

The Planets

on Dragon

The Dragon 32 cannot put text on to the high resolution graphics screen on its own but with a bit of thought it is possible. There are many methods of doing this, but the one which I use is this.

In Mode 3, the 4-colour mode, the memory map is arranged so that 1 byte covers 4 points on the screen i.e. each point takes 2 bits. Thus in mode 3 4 yellow points have the arrangement of

128	64	32	16	8	4	2	1
0	1	0	1	0	1	0	1

for any particular byte.

For a green point the code is	00
For a yellow point the code is	01
For a blue point the code is	10
For a red point the code is	11

To design a character you take a 4 × 5 matrix and fill in the points e.g. an A is

128 64 32 16 8 4 2 1

This in yellow, on a green background

would be coded 4 17 21 17 17

The screen is 32 bytes wide by 192 bytes deep in both modes 3 and 4 The address of the top left corner is 1536 irrespective of mode

The program The Planets uses this method to title the display The program is a simulation of four planets going around a central sun The planets are yellow on a green background As they go around they leave a red trace behind them The solar system is angled at about 30° from horizontal so the planets go around in ellipses of 2 1 length breadth ratio The animation is titled by the above method

Line 6 sets up the graphics screen to mode 3 with red blue yellow and green available with yellow as the foreground colour and red as the background colour The screen is cleared and set to green (the border colour) Line 8 draws the star in yellow Lines 10 to 14 draw the title and underline it Text is also in yellow

Lines 16 to 25 plot and unplot the planets, using the general formula for the circle of (cos(θ) sin(θ)) Line 26 sends the flow back to the start of the planet moving sequence so that the planets do not stop Lines 28 to 34 provide the Data for the title which reads

```
1 REM **THE PLANETS**
2 REM
3 REM (c) M.layley 1982
4 REM
5 REM **set hi-res screen
6 PMODE 3:SCREEN 1,1:COLOR 2,4:PCLS1
7 REM **draw star
8 CIRCLE (127,95),5,2
9 REM **print title
10 FOR I=6825 TO 7017 STEP 32
11 FOR J=0 TO 10
12 READ B
13 POKE I+J,B
14 NEXT J,I
15 REM **move planets
16 FOR I=0 TO -6.24 STEP -0.2
17 PSET (127+20*COS(I*8),95+10*SIN(I*8),2)
18 PSET (127+40*COS(I*4),95+20*SIN(I*4),2)
19 PSET (127+80*COS(I*2),95+40*SIN(I*2),2)
20 PSET (127+120*COS(I),95+60*SIN(I),2)
21 PRESET (127+20*COS(I*8),95+10*SIN(I*8))
22 PRESET (127+40*COS(I*4),95+20*SIN(I*4))
23 PRESET (127+80*COS(I*2),95+40*SIN(I*2))
24 PRESET (127+120*COS(I),95+60*SIN(I))
25 NEXT I
25 GOTO 16

27 REM **data for title
28 DATA 21,17,21,0,20,16,4,17,21,21,21
29 DATA 4,17,16,0,17,16,17,21,16,4,16
30 DATA 4,21,20,0,20,16,21,21,20,2,21
31 DATA 4,17,16,0,16,16,17,21,16,4,1
32 DATA 4,17,21,0,16,21,17,17,21,4,21
33 DATA 0,0,0,0,0,0,0,0,0,0,0
34 DATA 21,85,85,85,85,85,85,85,85,85,85
35 END
```

The Planets
by Martin Layley

Better than Basic

Can you program in a computer language other than Basic?

Enter this challenging new competition and win a Jupiter Ace.

Basic for all its advantages, is slow Programs written in Basic tend to look rather pedestrian when compared to programs written in some other languages such as machine code.
We want something different something faster than Basic it could be machine code, Forth Lisp Pascal or Fortran In fact, your entry can be written in anything that is not Basic And the best non-Basic program, be it game, utility or other will win the Jupiter Ace

The entries will be judged by *Popular Computing Weekly* editor, Brendon Gore, and Jupiter Ace designers Richard Altwasser and Steve Vickers In their selection account will be taken both of the standard of the program and of the accompanying documentation The whole range of languages and types of program are allowed The only stipulation is that it must not be written in Basic

Entries to the award scheme must be accompanied by four of the numbered coupons published in *Popular Computing Weekly* throughout October The closing date for the competition is November 16 The winning entry will be announced in the issue published on December 23

Rules

1. *There is no limit on the number of entries you can send in but each entry must be accompanied by four differently numbered competition coupons*
2. *Closing date for entries is November 16 1982*
3. *The names of the winners will be announced in the December 23 issue of Popular Computing Weekly*
4. *The Judges decision is final*
5. *No employees of Sunshine Publications Ltd or their families will be eligible to enter the competition*

Popular Computing Weekly
Better than Basic Competition

Fill in this coupon When you have collected four differently numbered coupons send them with your program to *Popular Computing Weekly* Better than Basic Hobhouse Court, 19 Whitcomb Street London WC2

NAME ....

ADDRESS ....

....

....

....

4

In this slot various contributors explore different aspects of the ZX Spectrum

Decorative line work on the small screen

Malcolm Davison holds up a mirror to the cosmos — with impressive results

For producing patterns on the screen, the *Draw* statement has a lot to offer. Here are a few examples.

The *Tent* program was carefully mapped out on a chart — an enlarged version of the one in the Sinclair manual (page 102) — before I commenced coding. Drawing a series of lines to meet another sloping line might have presented problems in establishing the *x* and *y* co-ordinates of their intersection. In fact, this was very straightforward.

Consider the *x* axis first. If 20 lines intersect one sloping line at equal intervals then both the *x* co-ordinates of the sloping line's ends may be subtracted and sub-divided into 20 as well, allowing you to establish the new *x* co-ordinates of each intersection. This may also be done for the *y* co-ordinates.

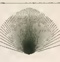
The *fan* which was superimposed over the tent posed an interesting problem as I needed equal lengths for each spoke. The *Draw* statement does not allow you to give the distance from the fixed starting point, so I had to resort to Pythagoras's Theorem (see line 170) to establish the *x* and *y* co-ordinates of the furthest end of the spoke.

The *fans* program started like the fan in the *tent* program. Changing the co-ordinates on the *Draw* statement to negative instead of positive and altering the *Plot* statement to a point at the top of the screen allowed two fans to be superimposed. Increasing the value of *z* by a smaller increment — *4 instead of *8 — increased the number of lines, giving a more effective interference pattern between the two fans.

Feather is another variation on the original fan — but the length of the arm is reproduced in steps (see line 170) by *c* for every spoke. *Peacock* is a further refinement on *feather*. The plot position is moved up the *y* axis for each spoke drawn — by the value of *c*.

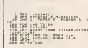
The idea behind *Cosmos* was straightforward — to produce mirror images of the basic fan program. However there was much juggling with the basic plot positions, range of values for *c* and length of the spoke until a neat pattern resulted within the bounds of the screen. But much of this was by trial and error altering the coding and running the program to see the effect.

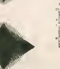
This idea of mirror imaging is very useful and quite easy to do. Produce a pattern anywhere on the screen — adjust its *x* and *y* co-ordinates to a more suitable place — then reproduce its mirror images. By putting the *Plot* and *Draw* paired statements in the correct sequence, the build up of the picture can look very effective indeed. A partly completed pattern — by pressing the *Break* and *Shift* keys — can be just as effective as the complete pattern.

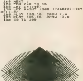


```
  1 REM  peacock
  2 REM program © M.Davison
 10 INK 2  PAPER 3  BORDER 0  C
LS
100 FOR c=0 TO 20
160 LET z=(c*4)
170 LET w=INT (SQR (((140 c)↑2)-(z↑2))+.5)
180 PLOT 125,18+c  DRAW z,w
185 PLOT 125,18+c  DRAW -z,w
190 NEXT c
200 GO TO 200
```

```
  1 REM  feather
  2 REM program © M.Davison
 10 INK 1  PAPER 3  BORDER 1  C
LS
100 FOR c=0 TO 20
160 LET z=(c*4)
170 LET w=INT (SQR (((145-c)↑2)-(z↑2))+.5)
180 PLOT 125,18  DRAW z,w
185 PLOT 125,18  DRAW -z,w
190 NEXT c
200 GO TO 200
```

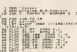






```
  1 REM  tent
  2 REM program © M.Davison
  5 BORDER 5  PAPER 4  INK 7  C
LS
 50 FOR c=0 TO 22
 60 LET x1=18+(c*4)
 70 LET x2=237-(c*4)
 80 PLOT x1,18  DRAW 85+c-(4*c),(INT (144/22*c)+.5)
 90 PLOT x2,18  DRAW ((c*4)-88-c),INT (144/22*c)+.5)
100 NEXT c
110 FOR c=1 TO 20
120 PLOT 125-c,31  DRAW -(20-2*c),(120/20*c)
130 PLOT 125+c,31  DRAW (20-2*c),(120/20*c)
140 NEXT c
150 FOR c=0 TO 18
160 LET z=(c*8)
170 LET w=INT (SQR ((145↑2)-(z↑2))+.5)
180 PLOT 125,18  DRAW z,w
185 PLOT 125,18  DRAW -z,w
190 GO TO 190
```

```
  1 REM  cosmos
  2 REM program © M.Davison
 10 INK 7  PAPER 2  BORDER 1  C
LS
 50 FOR c=0 TO 18
 60 LET z=(c*4)
 70 LET w=INT (SQR (((158-c)↑2)-(z↑2))+.5)
 80 PLOT 31,87  DRAW w,z
 82 PLOT 31,87  DRAW w,-z
 85 PLOT 125,158  DRAW z,-w
 86 PLOT 125,158  DRAW -z,-w
 90 PLOT 223,87  DRAW -w,z
 92 PLOT 223,87  DRAW -w,-z
 94 PLOT 125,18  DRAW -z,w
 96 PLOT 125,18  DRAW z,w
100 NEXT c
110 GO TO 110
```



```
  1 REM  fans
  2 REM program © M.Davison
 10 INK 7  PAPER 1  BORDER 1  C
LS
 50 FOR c=0 TO 38
 60 LET z=(c*4)
 70 LET w=INT (SQR ((160↑2)-(z↑2))+.5)
 80 PLOT 125,7  DRAW z,w
 82 PLOT 125,7  DRAW -z,w
 85 PLOT 125,167  DRAW z,-w
 88 PLOT 125,167  DRAW -z,-w
100 NEXT c
110 GO TO 110
```

Getting blood from a stone

John Durst's memory miser program shows how to get the most out of your Ram.

No matter how easy and cheap it may be computer memory is one commodity which you never seem to have enough of — but it's surprising how much of it can be found lying about in unswept corners of the Ram. The examples below are for the ZX81, but the principles apply to any computer.

Consider a block of data — a list of names, say — filed away in an array. Each letter is contained in a single byte as a number (the code). On the ZX81 for instance *A* is represented by 38 and *Z* by 63. But even *Z* the highest number in the alphabet set only uses six of the available eight bits in the byte.

In binary code 63 is 0011 1111. The two left-hand bits, which are needed to code numbers from 64 to 255 are not required. So they could be used for something else.

You may say that two bits does not sound very much of a saving but, remember, it is a quarter of every character. If you have a block of 200 characters you are wasting the equivalent of another 50 characters.

Inverse letter

The problem is how to get at those extra bits without much difficulty. Here is a simple program which uses bit No 7 — the leftmost bit — to code for the title (Mr Ms etc) on a name. It is easy to set bit No 7 to 1 on the ZX81. For a letter you just use the inverse letter. The code for an inverse video letter is the code for the letter plus 128. (Check it in the manual). One hundred and twenty-eight is represented in binary as 1000 0000. So *Z* would be 1011 1111.

Using this inverse letter system you could use the first two letters of a name to code for four different titles. Letters 1 and 2 inverse stand for "Ms", Letter 1 on and Letter 2 off stand for "Mrs". Letter 2 on and Letter 1 off stand for "Miss" and both Letters 1 and 2 off stand for "Mr".

As you see it's a simple matter to write the coded information, but how will you (or rather the computer) get it out again? One way would be to write a Basic program, to look at each letter in turn *For J= 1 To Len A$*. A quicker and neater way would be to use a couple of lines of machine code.

Figure one lists the machine code required for a ZX81. It will locate your name (provided you have it in *Z$*) examine the first two letters and return to Basic with the appropriate number from 0 to 3 depending on whether the first two letters of the name are inverse or not. It will also change the inverse letters into normal format.

If you are not entirely conversant with machine code, Figure two gives you a short program which will enter any machine code into a *Rem* statement in Line one.

Figure three shows you what happens to Line one when you *Run* the program. Once the program has been *Run* you can delete Lines 10 to 50 if you wish. Lines 100 to 130 are a test program to show you how to make the idea work. It will print out "Miss Smith". Try different combinations of inverse letters in the first two letters of "Smith" in Line 100 to get the other titles.

Remember to use this machine code program you must get your coded name into *Z$* because that is where the program expects to find it. If your names are stored in *A$* for example, you must include a line in your Basic program, such as *Let Z$ = $ (J)* immediately before the line with *Usr* 16514. This will also preserve the original coding for the title in A$ as Z$ will be altered by the machine code program so as to make the inverse letters normal.

The instruction at Code 4092 (in Figure 1) controls the number of letters examined and bits included in the code. If you alter 02 to 03 or 04 you will be able to code for 8 numbers or 16 numbers respectively. Looking at Figure 2 this means altering 0602 in the middle of the second line of *A$* to 0603 or 0604.

You can obviously push this system much further so as to use all the spare bits in your data. It is quite possible — and may well be worthwhile — to code for example, a 12-figure telephone number on top of a 24-letter name thereby saving up to 50 percent of your data space in the Ram.

CODING & DECODING FOR "MR", "MRS." ETC.

```
4082   3E 5F      LD   A,5F
4084   01 FC FF   LD   BC,FFFC
4087   2A 14 40   LD   HL,(4014)
408A   2B         DEC  HL
408B   03         INC  BC
408C   BE         CP   (HL)
408D   20 FB      JR   NZ,408A
408F   23         INC  HL
4090   23         INC  HL
4091   23         INC  HL
4092   06 02      LD   B,02
4094   AF         XOR  A
4095   CB 06      RLC  (HL)
4097   17         RLA
4098   CB 3E      SRL  (HL)
409A   23         INC  HL
409B   10 F8      DJNZ 4095
409D   4F         LD   C,A
409E   C9         RET
```

Fig 1

Decode No (0 to 3) and return in BC

```
  1 REM 1111111111111111111111111
111111
  10 LET A$="3E5F01FCFF2A14402B0
3BE20FB2323230602AFCB0617CB3E231
0F84FC9"
  20 FOR J=1 TO LEN A$/2
  30 LET X=J*2
  40 POKE 16513+J,CODE A$(X-1)*1
6+CODE A$(X)-476
  50 NEXT J
```

Fig 2

```
  1 REM Y?  UNPLOT COPY E=RNOF
 4 CLS 777  ACS  *ACS Y7( SAVE
?TAN
  10 LET A$="3E5F01FCFF2A14402B0
3BE20FB2323230602AFCB0617CB3E231
0F84FC9"
  20 FOR J=1 TO LEN A$/2
  30 LET X=J*2
  40 POKE 16513+J,CODE A$(X-1)*1
6+CODE A$(X)-476
  50 NEXT J
 100 LET Z$="SMITH"
 110 LET N$="MR.  MISS MRS. MS.
"
 120 LET N=5*USR 16514
 130 PRINT N$(N+1 TO N+5);Z$

MISS SMITH
```

Fig 3

Machine Code

Ian Stewart and Robin Jones present a new series for beginners

Worry about it later

We can set up the initial values we need for the loop (Machine Code October 21) by defining a new opcode *Hex* which just sets a word to a required value. It isn't really an opcode at all since it isn't equivalent to a machine instruction, so we call it a pseudo-operation. The whole program looks like this (ignore the number in the left- and right-hand margins for the moment)

Opcode	Hex
ADD	0
LD	1
ST	2
HLT	3
SUB	4
JP	5
JPZ	6
JPN	7
CALL	8
RET	9
XAI	A

We also need to know where the beginning of the program is. That's a more or less arbitrary decision, so let's assume it's at 020.

```
020           LD     BASE     1 033
021           XAI             A 000
022           LD     N1       1 030
023           ST     COUNT    2 032
024 LOOP      ADD    COUNT    0 032
025           STI             2 800
026           SUB    COUNT    4 032
027           SUB    N20      4 031
028           JPZ    OUT      6 047
029           LD     COUNT    1 032
02A           ADD    N1       0 030
02B           ST     COUNT    2 032
02C           XAI             A 000
02D           ADD    N1       0 030
02E           XAI             A 000
02F           JP     LOOP     5 024
030 N1        HEX    0001     0 001
031 N20       HEX    0014     0 014
032 COUNT     HEX    0000     0 000
033 BASE      HEX    0000     0 000
```

The only symbolic address which doesn't appear in the left-hand column and is therefore still unspecified, is *Out*. We'll worry about it later.

The form of the program we now have is written in what is known as *assembly code*. On modern sophisticated computers there will be an *assembler program* whose function is to convert this into real machine code for us.

Hand Assembly

Alas, neither our hypothetical machine nor the ZX81 has such a program. So we have to do the job by hand. We need a table of opcodes and their equivalent hex values.

If you have any machine code subroutines/tips/games, please send them to: Machine Code, *Popular Computing Weekly*, Hobhouse Court, 19 Whitcomb Street, London WC2 7HF.

at 020. Since each instruction occupies one word, we can write down the address of each instruction. You'll see that I've done this down the left-hand side of the program. Now we can replace the opcodes and addresses by their hex equivalents. For instance *Ld base* becomes 1 033 since Base is now identified as 033. The right-hand margin shows the complete code.

The only instruction which needs further comment is *Jpz Out*, which encodes as 6 047. Why should *Out* be at 047? It could be elsewhere, but 047 is the first location it can be at. The reason is that the array is occupying the space from 033 to 046 (twenty words) and we obviously don't want to go clumping around inside the program's data area.

The index Register

When the X register is in use, the real instruction is formed by adding the address field to the contents of the X-register. For instance, if the X register contains 400 then the instruction *Ldx 005* has the same effect as *Ld 405*.

We'll pinch another bit of the address field to indicate when indexing is in operation, so the LDX 005 instruction looks like this

```
| 0 0 0 1 | 0 | 1 | 0 0 0 0 0 0 0 1 0 1 |
  opcode    |   |        address
            |   index flag
            indirection flag
```

In hex, that's 1405. Actually, there's nothing you can do with indexing that you can't do with indirection. It's just that it will do arithmetic with addresses automatically instead of leaving the job to you.

Before we get into the Z80's architecture let's consider some of the difficulties of the processor we have just described.

First, the 4-bit operation code only allows 16 different instructions. (OK, we cheated a little by allowing the indirection and indexing flags to spill over into the address field, but that in turn means we have limited the address size, and therefore the maximum size of memory.) The Z80 has 694 instructions. To give each of them a separate bit pattern means that we need an 8-bit field (1 byte), and even then some fudging is needed.

Second, our imaginary machine uses memory in a rather careless way. Some of the instructions don't use the address field (*Hlt*, *Ldi*, *Sti*, for instance), so a sequence of such instructions wastes 10 bits in every word.

The Z80 gets over this problem by allowing different instructions to have different lengths. Some instructions have no address field and are just 1 byte long, while others have a 1 byte address field and are 2 bytes long. Still other instructions have a 2 byte address field for a total of 3 bytes, and there are even some which have 2 byte opcodes. This means that the *Pc* can't increment by 1 for every instruction executed. It has to increment by the length of the instruction.

Third, we always have to handle 16-bit words, which is inconvenient if we're dealing with characters (which normally occupy a byte each). So it would be nice to allow both 8 bit *and* 16 bit operations.

Fourth, the fact that there is only one general-purpose register (the A-register) can be annoying. It often means that intermediate results have to be stored temporarily back in memory while some other calculation is done. The Z80 has a number of general purpose registers.

Peek & poke

Peek your problems to our address. Ian Beardsmore will poke back an answer.

TURN OF THE SCREW

J Robinson of 221 Station Road Coleshills writes

Q **About 15 months ago I purchased a ZX81 with a 16K Ram pack. Since then it has been plagued with problems. When I plug in the Ram pack the screen rolls and the curser fades or darkens. I have cleaned the port with no luck. What is wrong?**

A Firstly do not plug in your Ram pack while the machine is on, only when it is off. Secondly, I think your problem may not be to do with the computer, but with your television. I would suggest that you try re-tuning it slightly.

THE AGE OF CONVENTION

John Hitchon of Foley Road East Streetly Sutton Coldfield writes

Q **I think that I have now reasonably mastered my BBC micro model A. But, I have one query. What is the *Tab* key for?**

A The *Tab* key, as opposed to the *Tab* function, is not used as a command or function as such. It generates the ASCII code 9, and is used for such things as word processing. Although used under program control, it is very similar to the *Tab* key on a conventional typewriter, in that it sets spaces and margins.

WORDSMITHING CONVENTIONS

Gavin Lawrence of Choan Drive Updean writes

Q **I was thinking about getting a Jupiter Ace computer, but I read that it only works in Black and White. What would happen if I tried to use it on a colour set? Also, in a review of the Ace you said that the Spectrum was not properly memory mapped. Is this so?**

A The Jupiter Ace should work on any standard television set, whether that set is black and white or colour. However, as it currently has no colour facility the display will always be in black and white, even on a colour set.

The Spectrum is not so much 'improperly' memory mapped, rather it is unconventionally memory mapped. The mapping is such that it is easier to use a command such as *Print At* rather than *Peek* or *Poke*. Indeed, the manual even suggests this.

Instead of running consecutively, the addresses run in lines of 32. Thus the first 32 addresses are the first line on the screen, then the ninth line on the screen, then the seventeenth line, and so on until the first block of eight lines is completed. The map then goes back to the second line and works down, tenth line, eighteenth line and so on.

Thus there are 256 addresses between the very first pixel and the one directly below it. When the first 64 lines have been done (ie eight characters down) the next batch of eight is started and mapped in the same way followed by the last batch of eight. It is easier to show this with a short program.

```
10 FOR N=16384 TO 22527
20 POKE N,143
30 NEXT N
```

This *Pokes* an inverse space into the addresses in the display file. You can follow the system of memory mapping from the way the character is built up in successive lines.

JOURNEYING DOWN UNDER

R Smith of Grange Road, Bishopsworth, Bristol, writes

Q **Please could you answer the following queries. I am emigrating to Australia later this year and am contemplating taking a computer with me, either a Sinclair or an Acorn. I might consider one of the other new models if they appear. Will a micro built in England work over there? I believe they use the same voltage and frequency, but that the PAL 625 network is in fact VHF. Also how would guarantees and service arrangements be affected?**

A I have been unable to obtain a list of world television standards, so I cannot tell you exactly what the Australian system is. If you are buying a micro to take over there, I would advise that you get a Sinclair as they have an established dealership. But, if you take a Spectrum rather than a ZX81, there is a chance that your Australian dealer will not touch it, because as yet there are no Spectrums going abroad.

I will give you the address of a dealer in Australia. The best thing you can do is contact him, but be careful to ensure that he guarantees any work on your computer. As it is not one of his own, he might adapt it for you, but not guarantee it. If he will not guarantee the work then you will have nothing to lose by getting any necessary changes done at a local shop.

The Australian dealer is:

Consolidated Marketing Corporation (Import) PTY Limited
86 Nicholson Street
Abbotsfield
Melbourne
Australia 3067
Tel Melbourne 419-3033

DECIMALISED SPOTS

S J Spruzen of Woburn Sands, Milton Keynes, writes

Q **I recently bought a ZX81 and I think I have discovered a bug in my Rom. On my ZX81 you can type and enter, without any syntax error coming up, the following lines.**

```
10 RUN (Full stop after Run)
20 IF A=B THEN (There is nothing after the Then statement)
30 LOAD       (four shifted Ps)
```

Also when *Running* this programme, my ZX81 shows up with a 0/0 report code. This also happens on my friend's ZX81. Please tell me why.

A I have in fact covered a similar situation some time ago. It is not a full stop after the *Run* but a decimal point. The computer is asked to go to line nothing point nothing. It interprets this as 0 and so goes to the first available line, which in this essence sends it straight back where it has just come from, so creating a loop. I tried this on a ZX81 and, despite leaving it in *Fast* for 15 minutes, I could not get the report code 0/0. To be honest, I do not know how that came about. How long did you leave the program *Running*?

If you bypass the first line and *Run* 20, you will get an error code. As you say, line 20 is incomplete. This fault has been removed on the Spectrum, and line 10 would give you an error. A good way of showing this effect is as follows:

```
10 PRINT
20 GOTO 10
```

Again, this program will not *Run* on the Spectrum, but it will *Run* on a ZX81 and, with a slightly different display, on the Vic20 as well.

A TOUCHING TRIBUTE TO UNCLE CLIVE

Vic Newton of Kidderminster, Worcestershire writes

Q **I have decided to sit out the present spate of wonder computers and see what develops in the next year. The ZX81 still fascinates me. The more I learn, the more I find to learn.**

I have seen an advertisement for PSS in Coventry who claim that its QSAVE can *Load/Save* 16K in just 26 seconds. It is also supposed to give the ZX81 a *Verify* statement. It costs just under £15. Do you have any further information on this?

A It is nice to see people reaffirming their faith in Uncle Clive's little black box of idiosyncrasies. I have had some letters from worried ZX81 owners who fear their machine is going to disappear. There are several hundred thousand of them in this country, so while they will take a back seat to the Spectrum they will be unlikely to fade out.

The QSAVE you mentioned is supposed to be very good. By the time I got to their stand at the last Microfair in the Horticultural hall they had sold out. They have promised to send one to use for review. At the time of writing this has yet to arrive, but I would suggest that you keep an eye on the review section over the next few weeks.

● *Stop agonising over that problem. Write to Ian Beardsmore, Peek and Poke, Popular Computing Weekly, Hobhouse Court, 19 Whitcomb Street, London WC2 7HF.*

Ian Beardsmore regrets that he cannot answer each question personally, so please do not enclose a SAE.

Classified

Computer Swap
01-930 3266

Free readers entries to buy or sell a computer. Ring 01-930 3266 and give us the details

Spectrums for sale

SPECTRUM 16K. Best offer for immediate delivery Tel Neal 01 942 5797

48K SPECTRUM with video amp and printer little used £200 Tel 01-958 8228 Peter Knight 58 Hillside Gardens Edgware Middlesex

SPECTRUM 48K Boxed complete plus 2 x printer demonstration tape and magazines £200 Tel 01 521 1058

ZX SPECTRUM 48K unused offers wanted Telephone 01 360 0458 after 7 pm

ZX81s for sale

16K ZX81 plus books and software £50 Peter Sandford 7 Moor Close Lancaster Tel 0524-62443 after 6 pm

ZX81 16K plus cassette and 4 books including machine code and cassette recorder Cost over £140 will accept £70 ono Telephone 01 590 2358

ZX81 16K with Protofs 2 keyboard various manuals £80 of software £100 one Tel 01-650 1365 after 6 pm

ZX81 plus 16K Ram keyboard and graphics chip £90 ono Telephone 0742 368677 after 5 pm

ZX81 16K with Kyde keyboard manual books and programs on cassette £80 Tel 0923 776260

ZX81 16K, includes Galaxoane Flight Simulation Star Trek and Centipede books worth £12 plus good condition £85 Tel 01 954 3848 after 5 pm

ZX81 16K. £50 Tel 0272 552778

ZX81 16K RAM. Stabilised power supply cassette software Sinclair built £80 ono Stevens Maidstone Tel 0622 28512

ZX81 16K + QS High resolution and character generator boards inverse video custom case professional keyboard + £200-worth of programs manuals Bargain at £156 (or will separate) Offers? Tel 0293 24504

ZX81 16K, plus mags books £50 Tel 01 567 3727 Mr Bell

ZX81 16K Full size typewriter keyboard with numeric key pad and case plus software leads and power supply £140 ono Delivery if within 20 miles of Cambridge Tel 076382 218 Mr Ison

ZX81 16K six months old £95 with £200 of software Tel Trowbridge 8867

ZX80 8K Rom 16K Ram plus tapes software and books £50 Tel 01 566 5381

ZX80 perfect working order complete with leads adapter manual £18 O Leeming 2 Laycroft Close Canterbury Tel Canterbury 60623

ZX81 16K leads and manual lots of books and magazines over £50 worth of software still under guarantee for only £80 Tel 041-638 6077

16K ZX81 as good as new still under guarantee £70 Michael Homewood 541 London Road Ditton Maidstone Kent

SINCLAIR ZX81 home Computer complete with leads and manual Very good condition £30 Tel 01-467 7155 anytime

Commodores for sale

VIC20 cassette 16K expansion machine code Rom chess Rom Alien Rom cassettes books cost £400 accept £275 ono Tel 0726 4246 any time

PET 3000 SERIES 32K Redo high resolution board (300AM20 x 200) built in sound output cassette deck and program £650 Tel Bradford 585288 evenings

32K, 4000 PET Disc drive and approximately £1 000 of software £1 000 (or will separate) Tel 0524 405552 any time after 5 pm

VIC20 UNEXPANDED Joystick, Sargon 2 cartridge £40 of software £185 including Datacassette Tel 051 920 8188 after 7 pm

VIC20 plus cassette deck machine code monitor 16K on extra Ram plus software and books worth £800 £400 ono Tel 0638 665722

VIC20 11 5K Ram. Datassette 60 programs including Defender Amok Vicmen Scramble Carrage Super lander £250 ono Steven Bartlett Reading Tel 0734 84194 anytime

UNEXPANDED VIC20. Excellent condition tape deck software manual and book £190 Tel 0234 750072 Darren

VIC20 + CZN. Cassette unit + 2 books £220 Tel 0234 46394

VIC20 CZN cassette deck Jelly Monsters cartridge Space Invaders Amok cassettes £180 ono Tel 0906 813015

VIC20 cassette recorder super Expanded programs books etc. £188 Tel 021 777 8827

COMMODORE PET 8016 with cassette £350 ono Tel 01 759 1831

VIC20 plus cassette unit and super expander with software £195 ono Telephone 01 360 8875 after 5 pm

VIC20 plus C2N super expander plus avenger Super Lander and joystick with £30 of software £260 Phone 01 655 1402

VIC20 with accessories worth £800 will sell for £350 Telephone Portsmouth (0705) 833830 after 4 30 pm

VIC20 with cassette unit, 3 software tapes and 2 basic books £185 ono Must be seen Tel Middlesborough 246661

VIC20, cassette deck all Vic books utility cartridges 21K Ram expansion Graphics pack plus many games any offers? Mr S Derose 01 882 0082 (evenings + weekends)

VIC20 COMPUTER SYSTEM, plenty of software and cartridges worth over £830 will accept best offer Tel for details 0638 665722

VIC20 Super Lander games cartridge unwanted gift excellent condition, £15 Tel (0763) 46042

PET 8K. Complete with tape drive £200 ono Tel Plymouth 27721 days Plymouth 47876 evenings Mr Ingram

PET 4008, new Rom cassette deck 8K dust cover manuals books games and original packing £400 ono N Leach 01 363 8601 (evenings)

VIC20 plus cassette deck plus 3K Ram plus Road Race cartridge machine language monitor various arcade games Intro to Basic and other VIC books Price negotiable Tel 01-882 0083 after 5 pm and weekends

Acorns for sale

BBC MODEL A 32K [illegible] Tel [illegible] 570385

Compukits for sale

8K UK101 cased with Basic 5 Bear plus tool kit code kit and coder in E prom sound board 2x via £60 software expansion cass and tape recorder etc £200 Tel Billericay 55678

UK 101 wooden case with built in cassette plus extras £50 A Pettit Tel Glynde 452

Nascoms for sale

NASCOM 2 with Zeap N [illegible] Graphics Chip Ram-B card containing 48K of memory and power supply Tel 01 546 9783 after 8 pm

Tandys for sale

TRS80 16K Level [illegible] numeric print out all these plus manual plus over £150 of software games plus business programs, £150 ono Tel West Midlands 632792

TANDY MODEL 3 Level I 4K Ram tape recorder (built in screen) £200 ono Tim Gan 01-588 0666 (weekends only)

TRS80, Model 1 Level 2 16K, with VDU cassette player manuals teaching software games and books £300 Oxford 0865 724182 evenings

TRS-80 LEVEL II 16K with monitor books manuals and £90 of software Offers around £225 Tel 0742 301310 6 to 8 pm

For sale

FOR SALE the most sophisticated graphics in town Tectronics 4052 graphic computing system and 4631 hard copy unit including tapes manuals, etc Bargain at £4 350 for quick sale Telephone 01 360 0152.

APPLE II 48K plus 9in B+W monitor plus 5¼ disk drive controller £830 Tel 0734 781055 (evenings)

OVC MICRO COMPUTER Model A 32K Ram leads plus software £340 Maidstone 681827

SHARP MZ80K (48K) with manual and games software £350 Tel 0658 65884 Mr J F Gilbert after 6 pm

TEXAS TI/994 as new 3 modules and cassette software £115 ono Hargood Kettering Tel 0536 518440 evenings

ZX PRINTER plus 5 extra rolls of paper complete with additional PSU £55 Tel 01 272 3310 evenings only Mr Ferris

ACETRONIC Computer Game with Space Invaders plus three other cartridges £80 Tel Tilbury 77329 (evenings only)

Wanted

WANTED Vic20 and 16K Ram + cassette deck Willing to pay around £150 ono Telephone 01-575 0073 after 6 pm

SWAP VIC20 CARTRIDGES Spiders of Mars Sargon chess for adventure cartridge Star Battle Tel 061 368 6935 after 8 pm

WANTED — 48K SPECTRUM Telephone Mr Herbert Grimsby 814457

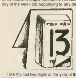
BBC MICROCOMPUTER model A expanded to 32K plus software Swap for Atari 400 system Ring Maidstone 681827

Rip van Winkle's awakening

by Gordon Lee

On the second of September 1752 the inhabitants of a certain village went to sleep and didn't awaken until September 14. Why?

While you are still trying to puzzle that one out let us look at a simple method for calculating the day of the week corresponding to any date.



Take the last two digits of the year of the date. Add to this number a quarter of the number disregarding any fraction. From the table below add the month value.

January	+1	(leap year +0)	July	+0
February	+4	(leap year +3)	August	+3
March	+4		September	+6
April	+0		October	+1
May	+2		November	+4
June	+5		December	+6

Now add the number of the day of the month and finally the century value from the table below.

2000 — 2099	6
1900 — 1999	0
1800 — 1899	2
14 9 1752 — 1799	4
1700 — 2 9 1752	1
1600 — 1699	2

To go back before 1900 just add 1 for each century you go back.

When you have a total divide by seven and check the remainder as follows 1 = Sunday 2 = Monday 3 = Tuesday 4 = Wednesday 5 = Thursday 6 = Friday 0 = Saturday. This gives us the day we are after.

For example consider October 21 1982. The last two digits of the year are 82 plus a quarter is 102 plus 1 for October plus 21 for the day of the month plus zero for the century which equals 124. Divided by 7 gives 17 with 5 left over. So the 21st is a Thursday.

The only problem is leap years. A year is a leap year if it is divisible by four eg 1960 1964 1968. But years that end in double zero — 1800 1900 etc although by rights leap years in fact are not except for the millennium years (2000 3000 etc) which are leap years!

Confused? If you are then consider the plight of the early Egyptians who found that their 365-day year slowly regressed with regard to the seasons and had to be periodically corrected. It was not until 46 BC that Julius Caesar added the extra day every fourth year to correct this.

All went fine until the middle of the 16th century when it was discovered that the equinoxes were occurring 10 days too late. In effect the Julian year was still 11 minutes a year too long which had a cumulative effect of eight days in every 1000 years.

Accordingly in 1577 Pope Gregory XIII introduced further reforms and declared that the century years should *not* be leap but that the millennium years should be. The only problem was the extra 10 days — so Pope Gregory issued instructions that the day after the 4th of October was to be 15th. However in Britain this system was not adopted until September 1752 when the 2nd of September was followed by the 14th which provides us with the answer to the riddle mentioned earlier.

For purists it should be added that the year is still too long by about 26 seconds so you may like to note in your diary that the year 4000 will *not* be a leap year!

Here is a program for working out the day on which any date fell this century. You may like to improve and adapt it to work for any date or perhaps to print out the calendar for any given month.

```
10 PRINT  ENTER YEAR
20 INPUT Y
30 IF Y  1900 OR Y  [illegible] THEN GOTO 20
40 PRINT  ENTER MONTH (JAN = 1 TO DEC
   12)
50 INPUT M
60 PRINT  NOW ENTER DATE
70 INPUT D
80 LET A$  SUNMONTUEWEDTHUFRISAT
90 LET Y$  STR$ Y
100 LET T  VAL Y$(3 TO 4)
110 LET T  INT (T + T 4)
120 IF M  1 OR M  10 THEN LET T  T + 1
130 IF M  2 OR M  3 OR M  11 THEN LET T  T
    + 4
140 IF M  5 THEN LET T  T + 2
150 IF M  6 THEN LET T  T + 5
160 IF M = 8 THEN LET T = T + 3
170 IF M  9 OR M  12 THEN LET T  T + 6
180 IF Y < > 1900 AND Y 4  INT (Y 4)  0 AND (M
    1 OR M  2) THEN LET T  T + 1
190 LET T  T + D
200 LET T  T  7
210 IF T < = 0 THEN GOTO 230
220 GOTO 200
230 LET T  T + 7
240 PRINT D   M   Y  IS WAS ON A  (A$(T
    *3  2 TO T *3)
```

Puzzle No. 28

What is the largest number that can be divided into each of the following four numbers to leave the same remainder in each case? The four numbers are 1702 3064 5334 and 6696.

Solution to Puzzle No 23

The solution uses the program we used to test for primes adapted to check *Z* for values of integers from 1 upwards. The program ends as soon as a non-prime is found.

```
10 LET Z = 1
20 LET T = Z * Z + Z + 41
30 FOR N  3 TO (SQR T) + 0 5
40 IF T N  INT (T N) = 0 THEN GOTO 100
50 NEXT N
60 LET Z  Z + 1
70 GOTO 20
100 PRINT T  ( N   T N )
```

The formula works for all values of *Z* from 1 to 39. However when $Z = 40$ we get the value 1681 which is equal to 41^2.

Winner of Puzzle No 23

The winner is R C Frost Barrowford Road Colne Lancashire who receives £10

A.R.T.H.U.R.

SHUT UP THE MAFIA ARE AFTER ME
ITALIANS EVERYWHERE AND I CAN PROVE IT

SIT DOWN RELAX HOW MANY BROTHERS HAVE YOU?
WHAT IS YOUR FATHER'S AGE AND OCCUPATION?
WHY HAVE YOU COME? WHAT CAN I DO TO HELP?

RELAX
HOW CAN I FEEL RELAXED IF YOU ASK QUESTIONS?

I ONLY WANT TO HELP YOU PLEASE SIT DOWN

HELP
NO ONE CAN HELP YOU ARE MY ENEMY
THE MAFIA ARE AFTER ME AND I
CAN PROVE IT YOU MUST BE ITALIAN TOO
WHY DO YOU ASK ME ALL THESE QUESTIONS? WHY?
AND IF YOU ASK ME WHY I ASK YOU WHY
I ASK YOU WHY YOU ASK ME WHY I ASK
INSERT A FOOTNOTE, I E CHUCKLE I
CAN GO ON LONGER I AM MADE OF METAL.

TELL ME ABOUT YOUR DREAMS LAST NIGHT FOR INSTANCE

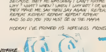
DREAMS
I DREAM ABOUT THE MAFIA FREQUENTLY
THE PROOF THEY'RE AFTER ME IS THAT THEY'LL GET ME
THEY STEAL MY LETTERS FROM THE LETTER BOX
THE PROOF IS THAT THE POSTMAN NEVER KNOCKS

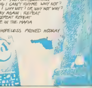
THEY FOLLOW ME AND WHEN I STOP THEY STOP
IF I GO SHOPPING THEN THEY'RE IN THE SHOP
THEY CAN CONTROL MY THOUGHTS, SO EVERY TIME
I WRITE A POEM THEY MAKE ME WRITE IN RHYME

TELL ME SOME MORE.

MORE
YES THERE IS MORE LOTS MORE AND I'VE GOT PROOF
ITALIANS EVERYWHERE AND I CAN PROVE IT
HOW CAN I FEEL RELAXED IF YOU ASK QUESTIONS
BECAUSE THEY'RE WATCHING I CAN'T RHYME WHY NOT?
WHY? WHAT? WHEN? WHICH? WHY NOT? OR WHY NOT WHY?
THEY MAKE ME SAY THEN SAY AGAIN REPEAT
REPEAT REPEAT REPEAT REPEAT REPEAT
AND SO DO YOU YOU MUST BE IN THE MAFIA

HOORAY I'VE PROVED IT'S HOPELESS PROVED HOORAY





COMPUTER SWAP 01-930 3266

Do you want to buy or sell a microcomputer? You can do it FREE in Computer Swap, a new regular service for *Popular Computing Weekly* readers

All you have to do is phone Computer Swap on 01-930 3266 and give us details of your computer, the price you want for it, your name, address and telephone number

Computer Swap entries are limited to a maximum of 30 words. They will be published in the first available issue

POPULAR Computing WEEKLY

CLASSIFIED ADVERTISING

Computer Swap — Free/Private reader — 10p a word/Trade Advertisement — 20p a word/Semi-display — £5 a single column centimetre, minimum two-column centimetres

Computer Swap — buy or sell your computer for free through Computer Swap. See box on left for details

Private readers — other advertisements from private readers cost 10p a word

Trade advertisements — cost 20p a word

Semi-display — why not make your advertisement more substantial by choosing the semi-display rate. It is only £5 a single column centimetre

Send your classified entries to Classified Department, *Popular Computing Weekly*, Hobhouse Court, 19 Whitcomb Street, London WC2. For semi-display enquiries call Alastair Macintosh on 01-930 3840

Here's my classified ad.

Please continue on a separate sheet of paper

I make this ______ words at ______ per word so I owe you £ ______

Name ______

Address ______

______

Telephone ______